GPC-Based Stable Reconfigurable Control

Don Soloway

Computational Sciences Division

NASA Ames Research Center

Moffett Field, CA 94035

Jianjun Shi and Atul Kelkar*

Department of Mechanical Engineering

Iowa state University

Ames, IA 50010

Abstract

This paper presents development of multi-input multi-output (MIMO) Generalized Predictive Control (GPC) law and its application to reconfigurable control design in the event of actuator saturation. A Controlled Auto-Regressive Integrating Moving Average (CARIMA) model is used to describe the plant dynamics. The control law is derived using input-output description of the system and is also related to the state-space form of the model. The stability of the GPC control law without reconfiguration is first established using Riccati-based approach and state-space formulation. A novel reconfiguration strategy is developed for the systems which have actuator redundancy and are faced with actuator saturation type failure. An elegant reconfigurable control design is presented with stability proof. Several numerical examples are presented to demonstrate the application of various results.

1 Introduction

Over last decade, GPC has emerged as one of the leading control design strategies for robust control of dynamical systems. In early years (prior to late eighties) GPC's applicability was essentially limited only to process control applications due to its demand on computational speeds. However, with the advances made in the computer technology over last decade computational speed is not a major concern for many real-life applications and control engineers have started using GPC for many main-stream applications. In recent years, GPC has become a viable alternative or in some cases even a preferred choice over well-known H_∞, H_2, and μ-synthesis approaches. GPC has proved to be very effective when requirements on the robustness and performance are hard to achieve with traditional control designs.

GPC belongs to a class of Model Predictive Control (MPC) methods. History of MPC dates back to late 70's when the process industry showed keen interest in using these control methods. The control formulation at the time was mainly heuristic and algorithmic [1, 2], and exploited the increasing potential of digital processors. These controllers were closely related to the minimum time optimal control methodology. The receding-horizon principle which is central to most of the MPC algorithms came about as early as 60's [3]. As mentioned earlier, MPCs became quite popular in the process industries where computational

speed was not a major concern. Also, many MPC algorithms were used on multivariable systems with constraints but no formal proofs of stability or robustness were available. Another parallel development took place using ideas from adaptive control which led to the development of self-tuning controllers [4] and extended horizon adaptive controllers (EHAC)[5]. This continued evolution of MPCs led to the emergence of the Generalized Predictive Control (GPC) methodology in late 80's [16] which incorporates all major features of the predictive controllers in a unified framework. Various versions of the same common idea gave rise to the following different types of predictive controllers: Multi-step Multivariable Adaptive Control (MUSMAR)[6], Multipredictor Receding Horizon Adaptive Control (MURHAC) [7], Predictive Functional Control (PFC) [8], and Unified Predictive Control (UPC)[9].

MPC has also been formulated in the state-space setting [10], which not only allows the use of well established state-space theories for analysis but also provides the ease for extensions to multivariable systems. Moreover, it facilitates the use of stochastic theories and treatment of actuator/sensor noise. The well developed estimation theory from state space methods can be easily incorporated without much complication. The perspective gained by working in these different domains made it possible to devise some simple tuning rules for ensuring stability and robustness for MPC systems. *As a simple analogy, MPC controller can be viewed as an observer-based controller wherein its stability, performance, and robustness is determined by the observer dynamics, which can be fixed by adjustable parameters, and regulator dynamics, determined by MPC parameters such as weightings, horizon lengths, etc.* In MPC, the control action at the current time instant is obtained by solving a finite horizon open-loop optimal control problem at *each* sampling instant. That is, the optimization problem yielding a control input is solved *on-line* at each time instant. The optimization process at each time instant yields an optimal control sequence and only the first control input in this sequence is applied to the plant. The novelty of MPC lies in its structure and not the control law itself. It essentially solves a standard nonlinear (or linear) optimal control problem. The fact that MPC solves the problem on-line and that it has the ability to naturally and explicitly handle the input and output constraints makes it a popular and powerful control paradigm.

One major criticism of MPC methodology is that, except for some special cases, there is no general purpose theory which guarantees closed-loop stability of MPC. Although, in [11], some specific stability theorems are given for GPC using the state-space setting the general case stability results for GPC were lacking. Recently, in 90's, stability of GPC under end-point constraints was shown in [12, 13] where the equality constraints were imposed on the output after at the end of a finite horizon. A variation of this end-point constraint idea is used in this paper for establishing stability of Multi-Input Multi-Output (MIMO) reconfigurable control architecture using GPC. The stability proof uses Linear Quadratic Regulator (LQR) results and monotonicity properties of Riccati solutions. This paper also presents a clean and detailed formulation of GPC prediction equations and derivation for end-point constraint-based GPC control law for MIMO linear time-invariant (LTI) systems. A stable reconfiguration capability of the proposed GPC architecture is demonstrated using a simulation example.
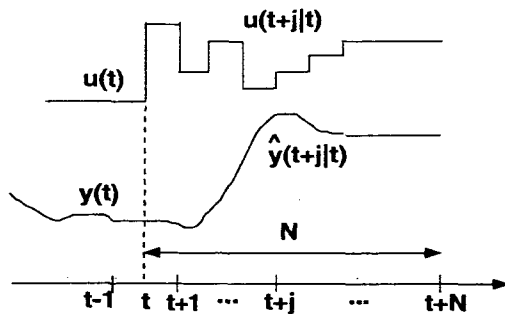
2

Figure 1: MPC Strategy

2 GPC Control Paradigm

GPC is the most generalized form of MPC. Among all different types of MPC schemes, GPC has maximum design freedom available for choosing design parameters. MPC design has several algorithms leading to different control schemes but all design schemes essentially use the same design paradigm. There are three basic elements of MPC: the prediction model, the cost function, and the control law. Every MPC uses some kind of plant model to predict future plant outputs over a pre-defined prediction horizon, $N_y = N_2 - N_1$, where, N_1 and N_2 represent lower and upper prediction horizons, respectively. The predicted outputs, $y(t + k|t)$ ($k = N_1, \ldots, N_2$) depend on the past inputs and outputs and on the future control signals $u(t + k|t)$, $k = 1, 2, \ldots, N_u$, where N_u represents a control horizon. A set of future control signals is calculated by optimizing a suitable performance index so as to keep the plant output as close to the reference trajectory $w(t + k)$ as possible. The performance index is typically a quadratic function of the predicted tracking error and control increments. The optimization is performed at each time step and the first element of the optimized control sequence is sent to the plant. The whole process is repeated again at the next time step after "receding" the prediction horizon by one time-step. Because of this *receding horizon* feature of this control scheme it is sometimes referred to as *Receding Horizon Control* as well. The MPC control strategy is illustrated in Figure 1. Figure shows the predicted output based on plant model and predicted optimal control sequence based on predicted future error. In the figure, the control horizon N_u is shown to be same as the prediction horizon N_y, which is not necessary. In case of GPC, all design parameters, N_1, N_2, N_u, and λ (control penalty in cost function) can be changed unlike other MPC schemes where one or more of these parameters are fixed. From this point on in the paper our focus will be on GPC control strategy unless otherwise mentioned. The control system block diagram for GPC is shown in Figure 2.

3 GPC Control of SISO Systems

Consider a discrete-time model of the single-input single-output (SISO) system described using a backward shift operator (q^{-1}) as,

$$A(q^{-1})y(k) = q^{-d}B(q^{-1})u(k - 1) + C(q^{-1})\frac{\xi(k)}{\Delta} \tag{1}$$
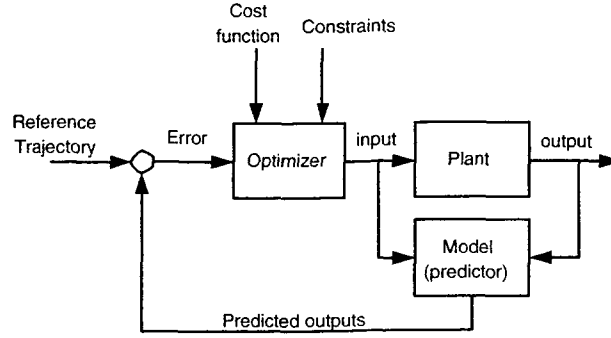
Figure 2: MPC block diagram

where $u(k)$ and $y(k)$ are the control and output sequences of the plant and $\xi(k)$ is an uncorrelated random sequence. A, B, and C are the polynomials in the backward shift operator q^{-1}:

$$
\begin{aligned}
A(q^{-1}) &= 1 + a_1 q^{-1} + a_2 q^{-2} + \cdots + a_{na} q^{-na} \\
B(q^{-1}) &= b_0 + b_1 q^{-1} + b_2 q^{-2} + \cdots + b_{nb} q^{-nb} \\
C(q^{-1}) &= 1 + c_1 q^{-1} + c_2 q^{-2} + \cdots + c_{nc} q^{-nc}
\end{aligned}
$$

where d is the dead time of the system. Δ is the difference operator $1 - q^{-1}$. This model is known as a Controlled Auto-Regressive Integrating Moving Average (CARIMA) model. For simplicity in the development, without loss of generality, $C(q^{-1})$ is chosen to be 1. The plant (1) is then given by:

$$
A(q^{-1})y(k) = q^{-d}B(q^{-1})u(k-1) + \frac{\xi(k)}{\Delta} \tag{2}
$$

The basic cost function used in GPC has the form

$$
J(N_1, N_2, N_u, \lambda) = \sum_{j=N_1}^{N_2} [\widehat{y}(k+j) - w(k+j)]^2 + \sum_{j=1}^{N_u} \lambda(j)[\Delta u(k+j-1)]^2 \tag{3}
$$

where, $\widehat{y}(k+j)$ is an optimum j-step ahead prediction of the system output up to time k, $w(k+j)$, $j = 1, 2, \ldots$ is a future set-point or reference sequence. N_1 is the minimum prediction horizon, N_2 is the maximum prediction horizon, N_u is the control horizon and $\lambda(k)$ is a control-weighting sequence. As seen in Eq. (3), the cost function is quadratic and penalizes future tracking errors over prediction horizon and control energy over control horizon. It is assumed that the control increments after control horizon are zero. The basic idea is to compute the *optimal* future control sequence, $\overline{u}^*(k) = [u(k), u(k+1), \ldots]$, such that the cost function $J(N_1, N_2, N_u, \lambda)$ is minimized. The control designer has to select the tuning parameters, N_1, N_2, N_u, and λ, to meet certain stability and performance objectives. Once $\overline{u}^*(k)$ is computed, only the first element of the sequence is used and the whole process is repeated at next time step.

3.1 Prediction of future outputs

In order to compute the cost function, which consists of future tracking errors, we need to compute the future outputs, $\widehat{y}(k+j)$ $(N_1 \leq j \leq N_2)$, using the best available plant

model (2). In case of SISO systems, these predictions can be easily computed using the Diophantine equation [9]. The development of prediction equations for SISO case using Diophantine approach is detailed in the Appendix 3.1. The prediction equations obtained in section 3.1 using Diophantine approach are very cumbersome to use and are not easily extendable to MIMO case. This motivates the development of state-space based formulation of prediction equations and control law.

3.2   State-space formulation of prediction equations

Consider a state-space description of the plant (Eq. 1) given as:

$$
\begin{aligned}
\mathbf{x}(k+1) &= \mathbf{A}\mathbf{x}(k) + \mathbf{B}\Delta u(k) + \mathbf{B}_\xi \xi(k) \\
y(k) &= \mathbf{C}\mathbf{x}(k) + \xi(k)
\end{aligned}
\tag{4}
$$

The reason for using the state-space form of the system with $\Delta u(k)$ as input is that the algebraic complexity in the derivation of GPC control law reduces significantly in this form. The transformation of system equations with $u(k)$ input to this form is given in detail in the appendix. In Eq. (4), the dimension of state vector is $n = max(n_a + 1, n_b + d + 1, nc)$, and matrices $\mathbf{A}$, $\mathbf{B}$, $\mathbf{B}_\xi$, and $\mathbf{C}$ are given by

$$
\mathbf{A} = \begin{bmatrix}
-\tilde{a}_1 & 1 & 0 & \cdots & 0 \\
-\tilde{a}_2 & 0 & 1 & \cdots & 0 \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
-\tilde{a}_{d+1} & 0 & 0 & \cdots & 0 \\
\vdots & \vdots & \vdots & \vdots & \vdots \\
-\tilde{a}_n & 0 & 0 & \cdots & 0 \\
\vdots & \vdots & \vdots & \vdots & \vdots \\
-\tilde{a}_{na+1} & 0 & 0 & \cdots & 0
\end{bmatrix}
\qquad
\mathbf{B} = \begin{bmatrix}
0 \\
0 \\
\vdots \\
b_0 \\
\vdots \\
b_{n-d-1} \\
\vdots \\
b_{nb-d-1}
\end{bmatrix}
$$

$$
\mathbf{B}_\xi = \begin{bmatrix}
c_1 - \tilde{a}_1 \\
c_2 - \tilde{a}_2 \\
\vdots \\
c_{d+1} - \tilde{a}_{d+1} \\
\vdots \\
c_n - \tilde{a}_n \\
\vdots \\
c_{nc} - \tilde{a}_{nc}
\end{bmatrix}
\qquad
\mathbf{C} = \begin{bmatrix} 1 & 0 & \cdots & 0 & \cdots & 0 & \cdots \end{bmatrix}
$$

where $\tilde{a}_i$ are the coefficients of polynomial $\tilde{A}$, which is given by

$$
\tilde{A}(q^{-1}) = \Delta A(q^{-1}) = 1 + \tilde{a}_1 q^{-1} + \cdots + \tilde{a}_n q^{-n} + \cdots + \tilde{a}_{na+1} q^{-(na+1)}
\tag{5}
$$

Note that, in matrix $\mathbf{B}$, the d leading elements are zeros. The block diagram representation of CARIMA model is shown in Fig. 3. If $d = 0$, the CARIMA model representation becomes that of Fig. 4. If in addition to $d = 0$, $C(q^{-1}) = 1$, the CARIMA model takes the form of Fig. 5, For both cases of Figs. 4 and 5, we can get the corresponding state-space
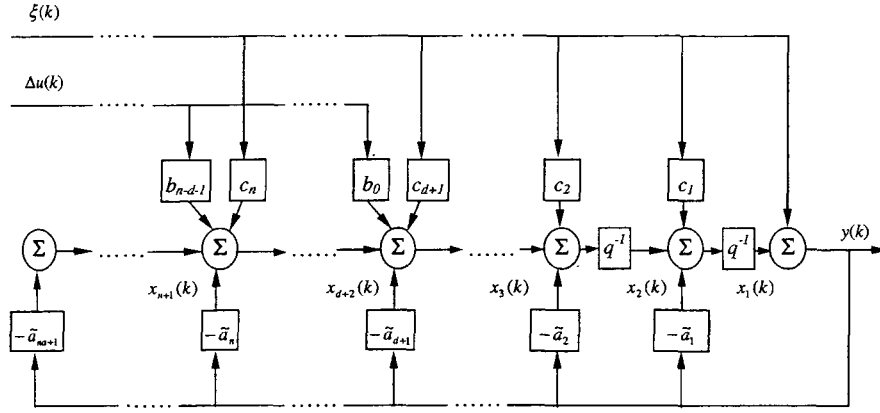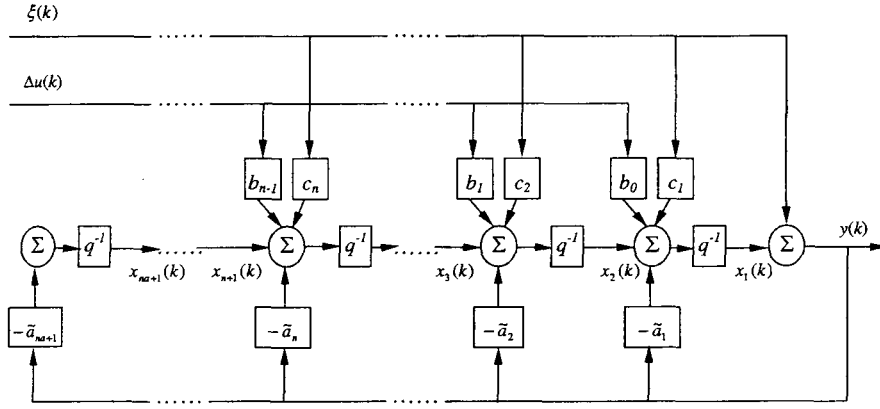
Figure 3: The CARIMA model

Figure 4: The CARIMA model with $d = 0$

model.

As mentioned previously, since the noise and disturbances in future are not known apriori for prediction of future outputs only deterministic part of the plant model is considered in the following development. The deterministic part of plant dynamics in polynomial description is given by

$$A(q^{-1})\Delta y(k) = B(q^{-1})\Delta u(k-1) \tag{6}$$

Let the corresponding state-space model be

$$\begin{aligned}
\mathbf{x}(k+1) &- \mathbf{Ax}(k) + \mathbf{B}\Delta u(k) \\
y(k) &= \mathbf{Cx}(k)
\end{aligned} \tag{7}$$

Using the z-transform, we can obtain the z-domain transfer function $\mathbf{H}(z)$ as follows:

$$\begin{aligned}
\mathbf{H}(z) &= \mathbf{C}(z\mathbf{I} - \mathbf{A})^{-1}\mathbf{B} \\
&= \frac{\mathbf{C}(\mathbf{I} - \frac{\mathbf{A}}{z})^{-1}\mathbf{B}}{z}
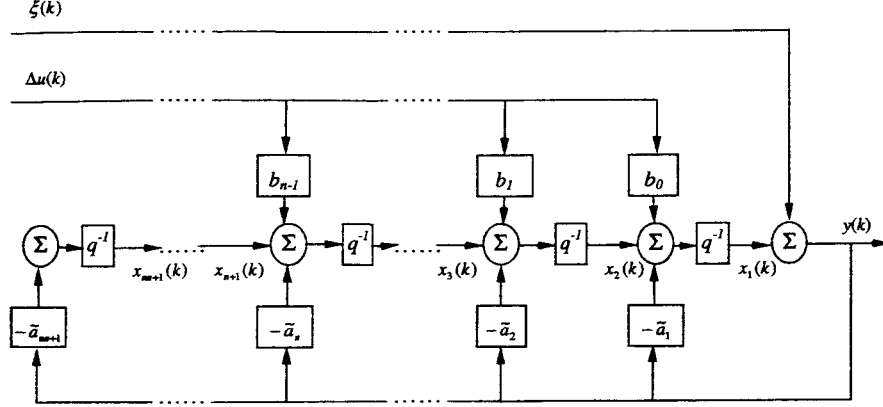\end{aligned}$$

Figure 5: The CARIMA model with $d = 0$ and $C(q^{-1}) = 1$

$$= \frac{\mathbf{C}}{z}\left(\mathbf{I} + \frac{\mathbf{A}}{z} + \frac{\mathbf{A}^2}{z^2} + \frac{\mathbf{A}^3}{z^3} + \cdots\right)\mathbf{B}$$

$$= \frac{\mathbf{CB}}{z} + \frac{\mathbf{CAB}}{z^2} + \frac{\mathbf{CA}^2\mathbf{B}}{z^3} + \cdots$$

By taking the inverse z-transform of the above equation, we get:

$$y(k) = \mathbf{CB}\Delta u(k-1) + \mathbf{CAB}\Delta u(k-2) + \mathbf{CA}^2\mathbf{B}\Delta u(k-3) + \cdots$$

The future outputs are then given by

$$y(k+1) = \mathbf{CB}\Delta u(k) + \mathbf{CAB}\Delta u(k-1) + \mathbf{CA}^2\mathbf{B}\Delta u(k-2) + \cdots$$
$$y(k+2) = \mathbf{CB}\Delta u(k+1) + \mathbf{CAB}\Delta u(k) + \mathbf{CA}^2\mathbf{B}\Delta u(k-1) + \cdots$$
$$\vdots$$
$$y(k+N_1) = \mathbf{CB}\Delta u(k+N_1-1) + \mathbf{CAB}\Delta u(k+N_1-2) + \mathbf{CA}^2\mathbf{B}\Delta u(k+N_1-3) + \cdots$$
$$\vdots$$
$$y(k+N_2) = \mathbf{CB}\Delta u(k+N_2-1) + \mathbf{CAB}\Delta u(k+N_2-2) + \mathbf{CA}^2\mathbf{B}\Delta u(k+N_2-3) + \cdots$$

A general term $y(k+j)$ $(j = 1, 2, \cdots, N_2)$ in the above equation can be written as

$$y(k+j) = \sum_{i=1}^{\infty} \mathbf{CA}^{i-1}\mathbf{B}\Delta u(k+j-i)$$

$$= \sum_{i=1}^{j} \mathbf{CA}^{i-1}\mathbf{B}\Delta u(k+j-i) + \sum_{i=j+1}^{\infty} \mathbf{CA}^{i-1}\mathbf{B}\Delta u(k+j-i)$$

$$= \sum_{i=1}^{j} \mathbf{CA}^{i-1}\mathbf{B}\Delta u(k+j-i) + \sum_{m=0}^{\infty} \mathbf{CA}^{m+j}\mathbf{B}\Delta u(k-m-1)$$

$$= \sum_{i=1}^{j} \mathbf{CA}^{i-1}\mathbf{B}\Delta u(k+j-i) + \sum_{m=0}^{\infty} \mathbf{CA}^{m+j}\left(\mathbf{x}(k-m) - \mathbf{A}\mathbf{x}(k-m-1)\right)$$

7

$$= \sum_{i=1}^{j} \mathbf{CA}^{i-1}\mathbf{B}\Delta u(k+j-i) + \sum_{m=0}^{\infty} \mathbf{CA}^{j}\left(\mathbf{A}^{m}\mathbf{x}(k-m) - \mathbf{A}^{m+1}\mathbf{x}(k-m-1)\right)$$

$$= \sum_{i=1}^{j} \mathbf{CA}^{i-1}\mathbf{B}\Delta u(k+j-i) + \mathbf{CA}^{j}\mathbf{x}(k)$$

Now using assumption of GPC control strategy that $\Delta u(k+i) = 0$ for $i \geq N_u$, and denoting

$$\widehat{\mathcal{Y}} = \begin{bmatrix} y(k+1) \\ y(k+2) \\ \vdots \\ y(k+N_1) \\ \vdots \\ y(k+N_2) \end{bmatrix} \quad \text{and} \quad \Delta\mathbf{u} = \begin{bmatrix} \Delta u(k) \\ \Delta u(k+1) \\ \vdots \\ \Delta u(k+N_u-1) \end{bmatrix}$$

we can write the predicted output as

$$\widehat{\mathcal{Y}} = \mathcal{G}\Delta\mathbf{u} + \mathcal{F}$$

where

$$\mathcal{F} = \begin{bmatrix} \mathbf{CA} \\ \mathbf{CA}^2 \\ \vdots \\ \mathbf{CA}^{N_1} \\ \vdots \\ \mathbf{CA}^{N_2} \end{bmatrix} \mathbf{x}(k)$$

and $\mathcal{G}$ is given by

$$\mathcal{G} = \begin{bmatrix} \mathbf{CB} & 0 & 0 & \cdots & 0 \\ \mathbf{CAB} & \mathbf{CB} & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \mathbf{CA}^{N_1-1}\mathbf{B} & \mathbf{CA}^{N_1-2}\mathbf{B} & \mathbf{CA}^{N_1-3}\mathbf{B} & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \mathbf{CA}^{N_2-1}\mathbf{B} & \mathbf{CA}^{N_2-2}\mathbf{B} & \mathbf{CA}^{N_2-3}\mathbf{B} & \cdots & \mathbf{CA}^{N_2-N_u}\mathbf{B} \end{bmatrix} \tag{8}$$

Note that Eq. (8) allows easy computation of $\mathbf{G}$ compared to Diophantine-based technique. For SISO systems, it is interesting to note that the polynomials E_j and F_j in Diophantine equation solution have coefficients that relate to the $\mathbf{A}$ matrix of the state-space form of Eq. (4). To be specific, the elements in the first row of $\mathbf{A}^j$ are the last N_u coefficients of E_{j+1} in the decreasing order of the degree of q^{-1} and the first column has the coefficients of F_j in the increasing order of the degree of q^{-1}.

3.3 Development of GPC for MIMO Systems

In this section, we will develop GPC control law for the general case of MIMO LTI systems. For algebraic simplicity we will use state-space description for plant dynamics. Let the

state-space model for a general MIMO system with p inputs and q outputs is given by

$$
\begin{aligned}
\mathbf{x}(k+1) &= \mathbf{A}\mathbf{x}(k) + \mathbf{B}\Delta\mathbf{u}(k) \\
\mathbf{y}(k) &= \mathbf{C}\mathbf{x}(k)
\end{aligned}
\tag{9}
$$

where, $\mathbf{x}$ is the n-dimensional state vector, $\mathbf{A}$, $\mathbf{B}$ and $\mathbf{C}$ are system matrices with dimensions $n \times n$, $n \times p$, and $q \times n$, respectively.

The cost function to be minimized is given by

$$
J(N_1, N_2, N_u, \lambda) = \sum_{j=N_1}^{N_2} (\mathbf{y}(k+j) - \mathbf{w}(k+j))^T (\mathbf{y}(k+j) - \mathbf{w}(k+j)) +
$$
$$
\sum_{j=1}^{N_u} \lambda(\Delta\mathbf{u}(k+j-1))^T (\Delta\mathbf{u}(k+j-1))
\tag{10}
$$

where, $\mathbf{y}$ is the q-vector of predicted output, $\mathbf{w}$ is the q-vector of reference trajectory, and $\Delta\mathbf{u}$ is the p-vector of input increments. In order to compute the optimal control input minimizing above performance function output predictions need to be computed over the prediction horizon. The following section develops prediction equations for MIMO case.

3.4 Output prediction for MIMO Systems

The procedure developed for predicting output in SISO case can be extended to MIMO systems as well with some added algebraic complexity. Consider a MIMO system of Eq. (9). We can compute the future outputs $\mathbf{y}(k+j)$ ($j = 1, 2, \cdots, N_1, \cdots, N_2$) based on the plant information available up to time k using a recursion procedure. The predictions of state for future times starting at time k are given by

$$
\begin{aligned}
\mathbf{x}(k+1) &= \mathbf{A}\mathbf{x}(k) + \mathbf{B}\Delta\mathbf{u}(k) \\
\mathbf{x}(k+2) &= \mathbf{A}\mathbf{x}(k+1) + \mathbf{B}\Delta\mathbf{u}(k+1) = \mathbf{A}^2\mathbf{x}(k) + \mathbf{A}\mathbf{B}\Delta\mathbf{u}(k) + \mathbf{B}\Delta\mathbf{u}(k+1) \\
\mathbf{x}(k+3) &= \mathbf{A}\mathbf{x}(k+2) + \mathbf{B}\Delta\mathbf{u}(k+2) = \mathbf{A}^3\mathbf{x}(k) + \mathbf{A}^2\mathbf{B}\Delta\mathbf{u}(k) + \mathbf{A}\mathbf{B}\Delta\mathbf{u}(k+1) \\
&\quad + \mathbf{B}\Delta\mathbf{u}(k+2) \\
&\vdots \\
\mathbf{x}(k+j) &= \mathbf{A}^j\mathbf{x}(k) + \sum_{i=0}^{j-1} \mathbf{A}^{j-i-1}\mathbf{B}\Delta\mathbf{u}(k+i)
\end{aligned}
$$

The output at time $k+j$, $\mathbf{y}(k+j)$, is then given by

$$
\mathbf{y}(k+j) = \mathbf{C}\mathbf{A}^j\mathbf{x}(k) + \sum_{i=0}^{j-1} \mathbf{C}\mathbf{A}^{j-i-1}\mathbf{B}\Delta\mathbf{u}(k+i)
\tag{11}
$$

Rewriting $\mathbf{y}(k+j)$ in the matrix form, we get

$$
\mathbf{y}(k+j) = \begin{bmatrix} \mathbf{C}\mathbf{A}^{j-1}\mathbf{B} & \mathbf{C}\mathbf{A}^{j-2}\mathbf{B} & \cdots & \mathbf{C}\mathbf{A}\mathbf{B} & \mathbf{C}\mathbf{B} \end{bmatrix} \begin{pmatrix} \Delta\mathbf{u}(k) \\ \Delta\mathbf{u}(k+1) \\ \vdots \\ \Delta\mathbf{u}(k+j-2) \\ \Delta\mathbf{u}(k+j-1) \end{pmatrix} + \mathbf{C}\mathbf{A}^j\mathbf{x}(k)
\tag{12}
$$

Let us define

$$\mathbf{y} = \begin{bmatrix} \mathbf{y}(k+N_1) \\ \vdots \\ \mathbf{y}(k+N_2) \end{bmatrix} = \begin{bmatrix} y_1(k+N_1) \\ y_2(k+N_1) \\ \vdots \\ y_q(k+N_1) \\ \vdots \\ y_1(k+N_2) \\ y_2(k+N_2) \\ \vdots \\ y_q(k+N_2) \end{bmatrix}_{(N_2-N_1+1)q \times 1} \tag{13}$$

and

$$\Delta\mathbf{u} = \begin{bmatrix} \Delta\mathbf{u}(k) \\ \Delta\mathbf{u}(k+1) \\ \vdots \\ \Delta\mathbf{u}(k+N_u-1) \end{bmatrix} = \begin{bmatrix} \Delta u_1(k) \\ \Delta u_2(k) \\ \vdots \\ \Delta u_p(k) \\ \Delta u_1(k+1) \\ \Delta u_2(k+1) \\ \vdots \\ \Delta u_p(k+1) \\ \vdots \\ \Delta u_1(k+N_u-1) \\ \Delta u_2(k+N_u-1) \\ \vdots \\ \Delta u_p(k+N_u-1) \end{bmatrix}_{N_u p \times 1} \tag{14}$$

Then predicted outputs are given in a compact form as

$$\mathbf{y} = \underbrace{\begin{bmatrix} \mathbf{CA}^{N_1-1}\mathbf{B} & \mathbf{CA}^{N_1-2}\mathbf{B} & \mathbf{CA}^{N_1-3}\mathbf{B} & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \mathbf{CA}^{N_2-1}\mathbf{B} & \mathbf{CA}^{N_2-2}\mathbf{B} & \mathbf{CA}^{N_2-3}\mathbf{B} & \cdots & \mathbf{CA}^{N_2-N_u}\mathbf{B} \end{bmatrix}}_{\mathbf{G}} \Delta\mathbf{u} + \underbrace{\begin{bmatrix} \mathbf{CA}^{N_1} \\ \vdots \\ \mathbf{CA}^{N_2} \end{bmatrix}}_{\mathbf{f}} \mathbf{x}(k)$$

$$\mathbf{y} = \mathbf{G}\Delta\mathbf{u} + \mathbf{f} \tag{15}$$

Note that the form of $\mathbf{G}$ in MIMO case is same as that of SISO case (Eq. (8)) except the prediction horizon is from N_1 to N_2. The cost function of Eq. (10) can be rewritten in a compact form as

$$\begin{aligned} J(N_1, N_2, N_u, \lambda) &= (\mathbf{y}-\mathbf{w})^T(\mathbf{y}-\mathbf{w}) + \lambda\Delta\mathbf{u}^T\Delta\mathbf{u} \\ &= (\mathbf{G}\Delta\mathbf{u}+\mathbf{f}-\mathbf{w})^T(\mathbf{G}\Delta\mathbf{u}+\mathbf{f}-\mathbf{w}) + \lambda\Delta\mathbf{u}^T\Delta\mathbf{u} \end{aligned} \tag{16}$$

10

where, $\mathbf{y}$, $\Delta\mathbf{u}$, $\mathbf{G}$, and $\mathbf{f}$ are given by Eqs. (13)-(15), and $\mathbf{w}$ is given by

$$\mathbf{w} = \begin{bmatrix} \mathbf{w}(k+N_1) \\ \mathbf{w}(k+N_1+1) \\ \vdots \\ \mathbf{w}(k+N_2) \end{bmatrix} = \begin{bmatrix} w_1(k+N_1) \\ w_2(k+N_1) \\ \vdots \\ w_q(k+N_1) \\ \vdots \\ w_1(k+N_2) \\ w_2(k+N_2) \\ \vdots \\ w_q(k+N_2) \end{bmatrix} \tag{17}$$

Rewriting the cost function as,

$$J = \frac{1}{2}\Delta\mathbf{u}^T\mathbf{H}\Delta\mathbf{u} + \mathbf{b}\Delta\mathbf{u} + \mathbf{f}_0 \tag{18}$$

where

$$\begin{aligned} \mathbf{H} &= 2[\mathbf{G}^T\mathbf{G} + \lambda\mathbf{I}] \\ \mathbf{b} &= 2[(\mathbf{f}-\mathbf{w})^T\mathbf{G}] \\ \mathbf{f}_0 &= (\mathbf{f}-\mathbf{w})^T(\mathbf{f}-\mathbf{w}) \end{aligned}$$

and setting

$$\frac{\partial J}{\partial \Delta\mathbf{u}} = 0$$

we obtain the optimal control input $\Delta\mathbf{u}^*$ as follows:

$$\Delta\mathbf{u}^* = -\mathbf{H}^{-1}\mathbf{b}^T = -(\mathbf{G}^T\mathbf{G}+\lambda\mathbf{I})^{-1}[\mathbf{G}^T(\mathbf{f}-\mathbf{w})] \tag{19}$$

The actual control signal sent to the process is the first p elements of the vector $\Delta\mathbf{u}^*$.

4 Stability of GPC control law

One of the major drawbacks of GPC control law presented in Eq. (19) is that the closed-loop stability under this control law can not be guaranteed. However, this difficulty can be overcome if the performance function in Eq. (10) used to obtain the control law can be modified to include the penalty on the end-state of the system and GPC design parameters can be chosen in a certain way.

The approach that is taken here to prove the stability of the GPC control law is to use the stability properties of the corresponding LQR problem. So, the optimal control problem can now be stated as follows:

For the LTI system

$$\begin{aligned} \mathbf{x}(k+1) &= \mathbf{A}\mathbf{x}(k) + \mathbf{B}\Delta\mathbf{u}(k) \\ y(k) &= \mathbf{C}\mathbf{x}(k) \end{aligned} \tag{20}$$

11

determine the optimal control law that minimizes the cost with the end-point state weighting

$$
\begin{aligned}
J(N_1, N_2, N_u, Q, \lambda) \;=\; & (\mathbf{x}(k+N_2) - \mathbf{w}_x(k+N_2))^T \mathbf{Q}(\mathbf{x}(k+N_2) - \mathbf{w}_x(k+N_2)) \\
& + \sum_{j=N_1}^{N_2} [\mathbf{y}(k+j) - \mathbf{w}(k+j)]^T [\mathbf{y}(k+j) - \mathbf{w}(k+j)] \\
& + \sum_{j=1}^{N_u} \lambda(j)[\Delta\mathbf{u}(k+j-1)]^T [\Delta\mathbf{u}(k+j-1)] \qquad (21)
\end{aligned}
$$

where, $\mathbf{w}_x(k+N_2)$ is the desired value of the state at the end of the prediction horizon, $N_1 = 1$, $N_2 = N_u = N$, $\mathbf{Q} \geq 0$ and $\lambda(j) = \lambda > 0$.

Since the stability properties of the system are independent of the reference input, we can ignore the reference input (or equivalently assume to be zero), i.e., set $\mathbf{w}_x(k+N_2) = 0$, and $\mathbf{w}(k+j) = 0$, $\forall j$. Then, the cost function will be modified to

$$
\begin{aligned}
J_N \;=\; & \mathbf{x}^T(k+N)\mathbf{Q}\mathbf{x}(k+N) \\
& + \sum_{j=1}^{N} \left\{ \mathbf{y}(k+j)^T \mathbf{y}(k+j) + \lambda \Delta\mathbf{u}(k+j-1)^T \mathbf{u}(k+j-1) \right\} \\
\;=\; & \mathbf{x}^T(k+N)(\mathbf{Q} + \mathbf{C}^T\mathbf{C})\mathbf{x}(k+N) \\
& + \sum_{j=0}^{N-1} \left\{ \mathbf{x}(k+j)^T \mathbf{C}^T\mathbf{C}\mathbf{x}(k+j) + \lambda \Delta\mathbf{u}(k+j)^T \Delta\mathbf{u}(k+j) \right\} \\
& - \mathbf{x}^T(k)\mathbf{C}^T\mathbf{C}\mathbf{x}(k) \qquad (22)
\end{aligned}
$$

4.1 GPC control law with end-point weighting

The GPC control law derived in Section 3.3 is no more optimal for the cost function in Eq. (22) since this cost function now includes additional penalty on the end-state of the prediction horizon which was absent in the cost of Eq. (16). The GPC control law for this modified cost is derived next.

The state evolution equation for the plant in Eq. (20) is given by

$$
\begin{aligned}
\mathbf{x}(k+N) \;=\; & \mathbf{A}^N\mathbf{x}(k) + \sum_{i=0}^{N-1} \mathbf{A}^{N-i-1}\mathbf{B}\Delta\mathbf{u}(k+i) \\
\;=\; & \mathbf{A}^N\mathbf{x}(k) + \underbrace{\left[\begin{array}{cccc} \mathbf{A}^{N-1}\mathbf{B} & \mathbf{A}^{N-2}\mathbf{B} & \cdots & \mathbf{B} \end{array}\right]}_{\overline{C}} \underbrace{\left(\begin{array}{c} \Delta\mathbf{u}(k) \\ \Delta\mathbf{u}(k+1) \\ \vdots \\ \Delta\mathbf{u}(k+N-1) \end{array}\right)}_{\Delta\mathbf{u}} \\
\;=\; & \mathbf{A}^N\mathbf{x}(k) + \overline{C}\Delta\mathbf{u} \qquad (23)
\end{aligned}
$$

Using Eqs. (23) and (15) the cost function in Eq. (22) can be re-written in more compact form as

$$
J_N = \frac{1}{2}\Delta\mathbf{u}^T \left[\mathbf{H} + 2\overline{C}^T\mathbf{Q}\overline{C}\right] \Delta\mathbf{u} + \left[\mathbf{b} + 2\mathbf{x}^T(k)(\mathbf{A}^N)^T\mathbf{Q}\overline{C}\right] \Delta\mathbf{u} + \mathbf{f}_0 + \mathbf{x}^T(k)(\mathbf{A}^N)^T\mathbf{Q}\mathbf{A}^N\mathbf{x}(k) \quad (24)
$$

where,

$$\begin{aligned}
\mathbf{H} &= 2\left[\mathbf{G}^T\mathbf{G} + \lambda\mathbf{I}\right] \\
\mathbf{b} &= 2\mathbf{f}^T\mathbf{G} \\
\mathbf{f}_0 &= \mathbf{f}^T\mathbf{f}
\end{aligned}$$

Matrices $\mathbf{G}$ and $\mathbf{f}$ are as defined in Eq. (15) and $\mathbf{f}$ is given as

$$\mathbf{f} = \underbrace{\begin{bmatrix} \mathbf{CA}^1 \\ \mathbf{CA}^2 \\ \vdots \\ \mathbf{CA}^N \end{bmatrix}}_{L}\mathbf{x}(k) = L\mathbf{x}(k)$$

Let the cost function in Eq. (24) be further simplified to

$$J_N = \frac{1}{2}\Delta\mathbf{u}^T\overline{\mathbf{H}}\Delta\mathbf{u} + \overline{\mathbf{b}}\Delta\mathbf{u} + \overline{\mathbf{f}}_0 \tag{25}$$

where

$$\begin{aligned}
\overline{\mathbf{H}} &= 2\left[\mathbf{G}^T\mathbf{G} + \lambda\mathbf{I} + \overline{C}^T\mathbf{Q}\overline{C}\right] \\
\overline{\mathbf{b}} &= 2\mathbf{x}^T(k)\left[L^T\mathbf{G} + (\mathbf{A}^N)^T\mathbf{Q}\overline{C}\right] \\
\overline{\mathbf{f}}_0 &= \mathbf{x}^T(k)\left(L^T L + (\mathbf{A}^N)^T\mathbf{Q}\mathbf{A}^N\right)\mathbf{x}(k)
\end{aligned}$$

Now the optimal control $\Delta\mathbf{u}^*$ can be obtained by setting the gradient of Eq. (25) to zero with respect to $\Delta\mathbf{u}$. That is,

$$\frac{\partial J}{\partial\Delta\mathbf{u}} = 0 \Rightarrow$$

$$\begin{aligned}
\Delta\mathbf{u}^* &= -\overline{\mathbf{H}}^{-1}\overline{\mathbf{b}}^T \\
&= -\left(\mathbf{G}^T\mathbf{G} + \lambda\mathbf{I} + \overline{C}^T\mathbf{Q}\overline{C}\right)^{-1}\left(\mathbf{G}^T L + \overline{C}^T\mathbf{Q}\mathbf{A}^N\right)\mathbf{x}(k) \\
&= -\mathbf{K}\mathbf{x}(k) \tag{26}
\end{aligned}$$

The following lemmas give the stability result for the corresponding LQR problem which will be used in proving the stability of GPC control law and the connection between GPC and LQR control laws.

Lemma 1 *For the system (20) the optimal control law minimizing the LQ performance index (22) is given by*

$$\Delta\boldsymbol{u}^*(k) = -K(k)\boldsymbol{x}(k) \tag{27}$$

where, gain $K(k)$ is given by

$$K(k+j) = [\boldsymbol{B}^T\widehat{P}(k+j+1)\boldsymbol{B} + \lambda\boldsymbol{I}]^{-1}\boldsymbol{B}^T\widehat{P}(k+j+1)\boldsymbol{A} \tag{28}$$

and $\widehat{P}$ is the solution of the following Riccati Difference Equation (RDE)

$$\widehat{P}(k+j) = \boldsymbol{A}^T\widehat{P}(k+j+1)\boldsymbol{A} - \boldsymbol{A}^T\widehat{P}(k+j+1)\boldsymbol{B}[\boldsymbol{B}^T\widehat{P}(k+j+1)\boldsymbol{B} + \lambda\boldsymbol{I}]^{-1}\boldsymbol{B}^T\widehat{P}(k+j+1)\boldsymbol{A} + \boldsymbol{C}^T\boldsymbol{C} \tag{29}$$

with the boundary condition

$$\widehat{P}(k+N) = \boldsymbol{Q} + \boldsymbol{C}^T\boldsymbol{C} \tag{30}$$

13

Proof 1 *Proof is given in the appendix.*

Lemma 2 *For the system shown in Eq. (20) and the performance function given in Eq. (22), the GPC control law given by Eq. (26) is same as the LQR control law (Eq. C.26) evaluated at the same time instant*

Proof 2 *Proof is given in the appendix.*

4.2 Stability result

The RDE in Eq. (C.28) can be re-written by reversing the time index by defining

$$P_m := \widehat{P}(k + N - m), \quad \text{where} \quad m = 0, 1, 2, \cdots, N - 1$$

Then, the RDE in the forward form becomes

$$P_{m+1} = \mathbf{A}^T P_m \mathbf{A} - \mathbf{A}^T P_m \mathbf{B}(\mathbf{B}^T P_m \mathbf{B} + \lambda \mathbf{I})^{-1} \mathbf{B}^T P_m \mathbf{A} + \mathbf{C}^T \mathbf{C} \tag{31}$$

with initial condition $P_0 = \mathbf{Q} + \mathbf{C}^T \mathbf{C}$.

Now, if we consider the steady-state solution for the infinite horizon problem, i.e., let $N \to \infty$ and let $P_m = P_{m+1} = P$ =constant as $m \to \infty$, then, Eq. (31) becomes the Algebraic Riccati Equation (ARE) in variable P:

$$P = \mathbf{A}^T P \mathbf{A} - \mathbf{A}^T P \mathbf{B}(\mathbf{B}^T P \mathbf{B} + \lambda \mathbf{I})^{-1} \mathbf{B}^T P \mathbf{A} + \mathbf{C}^T \mathbf{C} \tag{32}$$

The following lemma is state without proof which gives the conditions under which the matrix P has the stabilizing property.

Lemma 3 *If the system (A, B, C) is stabilizable and detectable, and $P_0 \succeq 0$, then the matrix sequence, $\{P_m\}_{m=0}^{\infty}$, generated by the RDE (Eq. (31)) has a unique limit P which stabilizes the system and satisfies the Algebraic Riccati Equation (ARE) (Eq. (32)).*

Theorem 1 *If the system given by Eq. (20) is stabilizable and detectable, $P_0 \succeq P_1$, $\mathbf{Q} \succeq 0$, $\lambda > 0$, $N_1 = 1$, and $N_2 = N_u = N$, then the GPC control law of Eq. (26) stabilizes the system.*

Proof 3 *In order to show stability of GPC control law, it suffices to show that the feedback gain K used in Eq. (26) is stabilizing. From Lemma 2, this is equivalent to showing that the corresponding LQR gain in Eq. (C.26) is stabilizing. Note that as long as the GPC parameters (N_1, N_2, N_u, and λ) are fixed the GPC control is equivalent to a constant-gain state feedback.*

Rewrite the RDE (Eq. (31)) as a set of two equations:

$$\begin{aligned} P_m &= A^T P_m A - A^T P_m B(B^T P_m B + \lambda I)^{-1} B^T P_m A + Q(m) \\ Q(m) : &= C^T C + (P_m - P_{m+1}) \end{aligned} \tag{33}$$

where, first of Eq. (33) is nothing but an ARE for a given m. They are also referred to as Fake Algebraic Riccati Equations (FARE) in the literature. Now, by Lemma 3 if the system is stabilizable and detectable and $P_m - P_{m+1} \succeq 0$, P_m will stabilize the system, i.e., $\overline{A} = A - B(B^T P_m B + \lambda I)^{-1} B^T P_m A$ will have all of its eigenvalues strictly within the unit circle. Next, we will use the monotonicity properties of the RDE and the property of FARE

14

to show the stability of GPC.

Consider matrix $\widetilde{P}_k$ defined as follows

$$\widetilde{P}_k = P_{k-1} - P_k$$

Then, from the result (Fact 5.2.1) on Page 95 of [22], $\widetilde{P}_k$ satisfies the RDE

$$\widetilde{P}_{k+1} = \widetilde{A}_k^T \widetilde{P}_k \widetilde{A}_k - \widetilde{A}_k^T \widetilde{P}_k B (B^T \widetilde{P}_k B + \widetilde{\lambda}_k)^{-1} B^T \widetilde{P}_k \widetilde{A}_k$$

where

$$
\begin{aligned}
\widetilde{A}_k &= A - B(B^T P_k B + \lambda I)^{-1} B^T P_k A \\
\widetilde{Q}(k) &= C^T C - C^T C = 0 \\
\widetilde{\lambda}_k &= B^T P_k B + \lambda I \succeq 0
\end{aligned}
$$

Now, if

$$\widetilde{P}_1 = P_0 - P_1 \succeq 0$$

by the property of the solution of the RDE,

$$\widetilde{P}_N = P_{N-1} - P_N \succeq 0.$$

Then, from first of Eq. (33), P_{N-1} stabilizes the system. Since, $P_{N-1} = \widehat{P}(k+1)$, the LQR controller gain $K(k)$ in Eq. (C.26) stabilizes the system. Now, from Lemma 2, GPC controller gain K in (Eq. (26)) stabilizes the system ■

4.3 GPC control law for non-zero reference trajectory

In the case when the desired trajectory is non-zero time varying the GPC control law (Eq. (26)) takes slightly different form to account for the tracking error. Consider the LTI system

$$
\begin{aligned}
\mathbf{x}(k+1) &= \mathbf{A}\mathbf{x}(k) + \mathbf{B}\Delta\mathbf{u}(k) \\
\mathbf{y}(k) &= \mathbf{C}\mathbf{x}(k)
\end{aligned}
$$

and the cost function

$$
\begin{aligned}
J(N_1, N_2, N_u, \lambda) = {}& (\mathbf{x}(k+N_2) - \mathbf{w}_x(k+N_2))^T \mathbf{Q}(\mathbf{x}(k+N_2) - \mathbf{w}_x(k+N_2)) \\
& + \sum_{j=N_1}^{N_2} [\mathbf{y}(k+j) - \mathbf{w}(k+j)]^T [\mathbf{y}(k+j) - \mathbf{w}(k+j)] \\
& + \sum_{j=1}^{N_u} \lambda(j)[\Delta\mathbf{u}(k+j-1)]^T [\Delta\mathbf{u}(k+j-1)] \qquad (34)
\end{aligned}
$$

where $\mathbf{w}_x(k+N_2))$ is the desired value of $\mathbf{x}$ at $t = k + N_2$.

Now, using the equations given in Section 15 the above cost function can be re-written as

$$
\begin{aligned}
J = {}& (\overline{C}\Delta\mathbf{u} + \mathbf{A}^{N_2}\mathbf{x}(k) - \mathbf{w}_x(k+N_2)))^T \mathbf{Q}(\overline{C}\Delta\mathbf{u} + \mathbf{A}^{N_2}\mathbf{x}(k) - \mathbf{w}_x(k+N_2))) \\
& + (\mathbf{y} - \mathbf{w})^T (\mathbf{y} - \mathbf{w}) + \lambda\Delta\mathbf{u}^T\Delta\mathbf{u}
\end{aligned}
$$

Substituting $\mathbf{y}$ by from Eq. (13), we can rewrite the cost function as

$$J = \frac{1}{2}\Delta\mathbf{u}^T\mathbf{H}\Delta\mathbf{u} + \mathbf{b}\Delta\mathbf{u} + \mathbf{f}_0$$

where

$$
\begin{aligned}
\mathbf{H} &= 2(\mathbf{G}^T\mathbf{G} + \lambda\mathbf{I} + \overline{C}\mathbf{Q}\overline{C}) \\
\mathbf{b} &= 2\left[(\mathbf{A}^{N_2}\mathbf{x}(k) - \mathbf{w}_x(k+N_2))^T\mathbf{Q}\overline{C} + (\mathbf{f} - \mathbf{w})^T\mathbf{G}\right] \\
\mathbf{f}_0 &= (\mathbf{A}^{N_2}\mathbf{x}(k) - \mathbf{w}_x(k+N_2))^T\mathbf{Q}(\mathbf{A}^{N_2}\mathbf{x}(k) - \mathbf{w}_x(k+N_2)) + (\mathbf{f} - \mathbf{w})^T(\mathbf{f} - \mathbf{w})
\end{aligned}
\tag{35}
$$

The optimal control law is then given by

$$\Delta\mathbf{u}^* = -\mathbf{H}^{-1}\mathbf{b}^T. \tag{36}$$

where $\mathbf{H}$ and $\mathbf{b}$ are given by Eq. (35).

4.4 Computation of desired state trajectories

In most real life situations it is safe to assume that the desired output trajectory is given to the control designer. However, the control law computation (Eq. 36) requires that the desired state trajectories are known. This section gives the procedure for computing the desired state trajectories given the desired output trajectories for a general case.

Consider MIMO LTI system given by the following state-space description

$$\mathbf{x}(k+1) = \mathbf{A}\mathbf{x}(k) + \mathbf{B}\Delta\mathbf{u}(k). \tag{37}$$

Let $\mathbf{w}(k+N_2)$ denote the desired output trajectory that needs to be tracked by the system and let $\mathbf{w}_x(k+N_2)$ denote the corresponding trajectories for the desired state. Then, $\mathbf{w}_x(k+N_2)$ must satisfy the following conditions:

$$
\begin{aligned}
\mathbf{w}(k+N_2) &= \mathbf{C}\mathbf{w}_x(k+N_2) \tag{38} \\
(\mathbf{A} - \mathbf{I})\mathbf{w}_x(k+N_2) &= 0 \tag{39}
\end{aligned}
$$

The first condition (Eq. 38) is a direct consequence of output equation. The second condition (Eq. 39) is derived next.

One consideration in tracking problem is that the choice of $\mathbf{w}_x(k+N_2)$ should be consistent with the zero steady-state output tracking error. Consider the GPC control law with tracking error feedback:

$$\Delta\mathbf{u}^*(k) = -\mathbf{K}e(k) \tag{40}$$

where

$$e(k) = \mathbf{x}(k) - \mathbf{w}_x(k)$$

The closed-loop error dynamics for the system given by Eq. (37) with control law (40) is given by

$$
\begin{aligned}
e(k+1) &= \mathbf{x}(k+1) - \mathbf{w}_x(k+1) \\
&= \mathbf{A}\mathbf{x}(k) + \mathbf{B}\Delta\mathbf{u}^*(k) - \mathbf{w}_x(k+1) \\
&= (\mathbf{A} - \mathbf{B}\mathbf{K})e(k) + \mathbf{A}\mathbf{w}_x(k) - \mathbf{w}_x(k+1).
\end{aligned}
$$

16

At steady-state, for zero tracking error we need

$$e(k+1) = e(k) = 0, \quad \text{and} \quad \mathbf{w}_x(k) = \mathbf{w}_x(k+1).$$

Then, to satisfy the end-point constraint of zero tracking error, we need

$$(\mathbf{A} - \mathbf{I})\mathbf{w}_x(k + N_2) = 0.$$

Combining the conditions of Eqs. (38) and (39), we get

$$\begin{bmatrix} \mathbf{A} - \mathbf{I} \\ \mathbf{C} \end{bmatrix} \mathbf{w}_x(k + N_2) = \begin{bmatrix} 0 \\ \mathbf{w}(k + N_2) \end{bmatrix} \tag{41}$$

Note that Eq. (41) typically represents under-determined system of equations as matrix on the left hand side is not generally invertible. In such as case, a least square solution can be obtained for the desired state trajectories as

$$\mathbf{w}_x(k + N_2) = \left(\begin{bmatrix} \mathbf{A} - \mathbf{I} \\ \mathbf{C} \end{bmatrix} \right)^{\dagger} \begin{bmatrix} 0 \\ \mathbf{w}(k + N_2) \end{bmatrix} \tag{42}$$

where $(\cdot)^{\dagger}$ denotes pseudo-inverse of $(\cdot)$.

4.4.1 Special Case: System with redundant actuators

For the system which has more inputs than outputs, i.e. $p > q$, the transformation of the system with $\mathbf{u}(k)$ as input to the form which has $\Delta\mathbf{u}(k)$ as input, the detectability may be lost. To see this, consider the matrix

$$\begin{bmatrix} \lambda\mathbf{I} - \mathbf{A} \\ \mathbf{C} \end{bmatrix} \tag{43}$$

where $\mathbf{A}$ and $\mathbf{C}$ are defined as in Eq. (B.16). When $\lambda = 1$, rank of $(\lambda\mathbf{I} - \mathbf{A})$ is less than $n - p$. The rank of C can at the most be q. So, the matrix in Eq. (43) must have rank less than n, which is not a full column rank. That is, the detectability is lost. However, this problem can be addressed as follows:

Consider a system with $\Delta\mathbf{u}(k)$ as input and let the number of inputs be greater than the number of outputs, i.e., $p > q$.

$$\begin{aligned} \mathbf{x}(k+1) &= \mathbf{A}\mathbf{x}(k) + \mathbf{B}\Delta\mathbf{u}(k) \\ \mathbf{y}(k) &= \mathbf{C}\mathbf{x}(k) \end{aligned} \tag{44}$$

Now consider a modified output $\widetilde{\mathbf{y}}(k)$ such that

$$\widetilde{\mathbf{y}}(k) = \widetilde{\mathbf{C}}\mathbf{x}(k) \tag{45}$$

where

$$\widetilde{\mathbf{y}}(k) = \begin{bmatrix} \mathbf{y}(k) \\ \mathbf{u}(k-1) \end{bmatrix}, \quad \widetilde{\mathbf{C}} = \begin{bmatrix} \mathbf{C} \\ \mathbf{0}_{p \times (n-p)} \quad \mathbf{I}_{p \times p} \end{bmatrix} \tag{46}$$

17

It can be verified that with respect to this new output the system is detectable. Thus, the modified system now takes the form:

$$\mathbf{x}(k+1) \;=\; \mathbf{A}\mathbf{x}(k) + \mathbf{B}\Delta\mathbf{u}(k) \qquad (47)$$

$$\tilde{\mathbf{y}}(k) \;=\; \tilde{\mathbf{C}}\mathbf{x}(k) \qquad (48)$$

which is stabilizable and detectable. For this new system, let the cost function be modified as follows:

$$\tilde{J} = \mathbf{x}^T(k+N)\mathbf{Q}\mathbf{x}(k+N) + \sum_{j=1}^{N}\mathbf{y}^T(k+j)\tilde{\mathbf{Q}}\mathbf{y}(k+j) + \sum_{j=1}^{N}\lambda\Delta\mathbf{u}(k+j-1) \qquad (49)$$

where

$$\tilde{\mathbf{Q}} = \begin{bmatrix} \mathbf{I}_{q\times q} & \\ & \mathbf{0}_{p\times p} \end{bmatrix} \qquad (50)$$

Now, it can be seen that minimizing the cost function given by Eq. (49) for the system given by Eq. (47) is same as minimizing the original cost function (Eq. 22) for original system (Eq. 44). The advantage in dealing with transformed system is that it satisfies the conditions of stability theorem and therefore allows computation of GPC control law.

Now, to show that the stability proof still holds consider the following:

We have already shown that the modified cost function is essentially same as the original cost function. Now it remains to show that the Riccati equations used in Lemma 1 with appropriate boundary condition do not change as a result of transformed system. The equations in Lemma 1 for the modified system can be rewritten as:

$$\Delta\mathbf{u}^*(k) = -\mathbf{K}(k)\mathbf{x}(k)$$

where

$$K(k+j) = [\mathbf{B}^T\widehat{P}(k+j+1)\mathbf{B} + \lambda\mathbf{I}]^{-1}\mathbf{B}^T\widehat{P}(k+j+1)\mathbf{A}$$

$$\widehat{P}(k+j) = \mathbf{A}^T\widehat{P}(k+j+1)[\mathbf{A} - \mathbf{B}K(k+j)] + \tilde{\mathbf{C}}^T\tilde{\mathbf{Q}}\tilde{\mathbf{C}}$$

with

$$\widehat{P}(k+N) = \mathbf{Q} + \tilde{\mathbf{C}}^T\tilde{\mathbf{Q}}\tilde{\mathbf{C}}.$$

By direct substitution, it can be easily verified that

$$\tilde{\mathbf{C}}^T\tilde{\mathbf{Q}}\tilde{\mathbf{C}} = \mathbf{C}^T\mathbf{C}.$$

This essentially proves that the GPC control law derived for the transformed system stabilizes the original system.

5 Reconfigurable control

In this section, we will extend the stability result obtained in previous section to the case of reconfigurable control architecture. We present a reconfigurable control scheme wherein the system has some redundancy in control actuators and in the event of actuator failures, like saturation for example, the system can reconfigure itself to re-allocate the control signal to other set of actuators.

Consider a strictly proper stabilizable and detectable MIMO LTI system from Eq. (9) with p inputs and q outputs.

$$
\begin{aligned}
\mathbf{x}(k+1) &= \mathbf{A}\mathbf{x}(k) + \mathbf{B}\Delta\mathbf{u}(k) \\
y(k) &= \mathbf{C}\mathbf{x}(k)
\end{aligned}
\tag{51}
$$

Let us suppose that the system has some redundancy in actuators, i. e., $p \geq q$ and some of the p inputs can control more than one outputs. The reconfiguration problem is defined as follows. If one or more of the p actuators saturate (i.e., become "inactive") how to re-allocate the control signal to actuators which are still "active" such that the lost degrees of freedom due to saturation in controlling certain outputs are regained via re-allocation. Moreover, to ensure that the overall stability of the system is maintained under such reconfiguration. Before reconfiguration scheme is presented and stability of such scheme is established some definitions are in order.

5.1 Definitions

Definition 1 *Reconfiguration matrix: Any $q \times p$ matrix with binary entries (0 or 1) is called as Reconfiguration Matrix, Q_{rc}, and is used to set the control priorities for each actuator.*

Q_{rc} is used to select which actuator is used to control a certain output. The size of the Q_{rc} matrix is q (number of outputs) by p (number of inputs). Each element of Q_{rc} is either 1 or 0 indicating weather a particular actuator is allowed to control a specific output or not. For example, let us take a 5-input 2-output system. An example Q_{rc} matrix could be

$$
Q_{rc} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 \end{bmatrix}
\tag{52}
$$

In this example of Q_{rc}, the first output can be controlled only by the second input, and the second output can be controlled by both the third and fourth inputs.

This matrix is then used to "reconfigure" the $\mathbf{H}$ matrix (where $\mathbf{H}$ is the transfer function matrix of system in Eq. (51)) as follows which causes the reconfiguration. Reconfigured $\mathbf{H}$ matrix is denoted by $\mathbf{H}_{rc}$ and is given by

$$
\mathbf{H}_{rc} = Q_{rc} \otimes \mathbf{H}
\tag{53}
$$

where, $\otimes$ indicates the outer product. This outer product effectively modifies the $\mathbf{B}$ and the $\mathbf{c}$ matrix of the system to account for reconfiguration.

Definition 2 *Let $\mathcal{Q}$ be the set of all allowable Q_{rc}s such that the outer product in Eq. (53) produces a set of reconfigured stabilizable and detectable systems.*

That is,
$$\mathcal{Q} = \left\{ Q_{rc}^i \mid (\mathbf{A}, \tilde{\mathbf{B}}_i, \tilde{\mathbf{C}}_i) \quad \text{is stabilizable and detectable} \quad \forall i \right\} \tag{54}$$

where, $\tilde{\mathbf{B}}_i$ is the reconfigured $\mathbf{B}$ matrix and $\tilde{\mathbf{C}}_i$ is the reconfigured $\mathbf{C}$ matrix as a result of outer product of Q_{rc}^i with $\mathbf{G}$.

Definition 3 *The matrix designed to prioritize the order of the actuators in which they are used in the event of saturation failure is called as Reconfiguration Order Matrix.*

The design of the reconfiguration matrix Q_{rc} is based on the criteria designed suitably based on the application, for example one of the criteria could be the saturation of the actuators. The order in which the redundant actuators are used is based on the reconfiguration order matrix, Q_{ro}. For example, for the Q_{rc} defined in Eq. (52) a reconfiguration order matrix could be

$$Q_{ro} = \begin{bmatrix} 4 & 1 & 2 & 2 & 3 \\ 4 & 3 & 1 & 1 & 2 \end{bmatrix} \tag{55}$$

This Q_{ro} matrix informs the reconfiguration algorithm to use the second input to control the first output until it saturates followed by actuators 3 and 4 until they saturate followed by actuators 5 and then 1. Similar logic is used for the second output. Each time the control law is evaluated, the first actuators are used to calculate the control input. The input is tested for saturation, if it is saturated, the Q_{rc} matrix is set to the next actuator (or set of actuators). This process is repeated until there is no change in the Q_{rc} matrix or there are no more free actuators left. Then, the resulting control is sent to the plant. This process is repeated at each cycle. The repeating of this process ensures that the reverse order of actuators is used as the actuators unsaturate.

5.2 Stability under reconfigurable control

The stability of the reconfigurable control architecture described above will be established in two steps. In the first step, it will be shown that the set of plants obtained after reconfiguration is stabilized by the GPC control law of Eq. (26). Then, in the second step, it will be shown that the system remains stable during the switching of the control law under reconfiguration. The stability of the reconfigured system is established by the following theorem.

Theorem 2 *All systems resulting from reconfiguration of system given by Eq. (51) due to reconfiguration matrix Q_{rc} are stabilized by the control law of Eq. (26) iff Q_{rc} belongs to the set $\mathcal{Q}$.*

Proof 4 *Proof of this theorem is the direct consequence of Definition 2 and Theorem 1.*

Note that the control law of Eq. (26) stabilizes the original system (Eq. (51)) as well according to Theorem 1. Having established that both systems - system before reconfiguration and the system after reconfiguration - are stabilized by the control law of Eq. (26), to complete the stability argument we need to show that the stability is retained during the process of reconfiguration. For that, let the system after reconfiguration is described by

$$
\begin{aligned}
\hat{\mathbf{x}}(k+1) &= \hat{\mathbf{A}}\hat{\mathbf{x}}(k) + \hat{\mathbf{B}}\Delta\mathbf{u}(k) \\
y(k) &= \hat{\mathbf{C}}\hat{\mathbf{x}}(k)
\end{aligned} \tag{56}
$$

20

Let us suppose that the system (51) is under closed-loop control with control law given by Eq. (26). Suppose the system has actuator saturation at time instant $k = r$ and is reconfigured by matrix $Q_{rc} \in \mathcal{Q}$ to yield the system of Eq. (56). That means, the dynamics of the system (51) starting at say initial time $k = 0$ is governed by the dynamics of system (56) after reconfiguration (i.e., for $\forall\, k \geq r$). Since the GPC control of Eq. (26) is stabilizing for system (51), $\mathbf{x}(r)$ is bounded. Let $\hat{\mathbf{x}}(r)$ be the initial condition for the system (56) at time instant $k = r$. Since $\hat{\mathbf{x}}(r)$ is related to $\mathbf{x}(r)$ by some similarity transformation, $\hat{\mathbf{x}}(r)$ is also bounded. Now, for time $k > r$, system evolves according the dynamics of Eq. (56). From Theorem 2, the system (56) is also stabilized by the control law of Eq. (26), i.e., $\hat{\mathbf{x}}(k)$ remains bounded for all $r < k \leq \infty$. Thus we have shown that the system remains stable for all $0 \leq k \leq \infty$ under control law of Eq. (26) $\blacksquare$

6 Numerical Example

In this section, a numerical example is given to demonstrate stable reconfiguration capability of the GPC control methodology presented in Section 5 using a short-period approximation model of an aircraft. Two different cases of reconfiguration are presented where each case can be considered to represent a different type of actuator limitation and/or failure.

The short-period approximation model of an aircraft considered is:

$$
\begin{aligned}
\dot{x}(t) &= \begin{bmatrix} 0 & -1.3677 \\ 1.0000 & -1.5087 \end{bmatrix} x(t) + \begin{bmatrix} -0.0665 & -0.0029 & -0.0029 \\ -0.0862 & -0.0043 & -0.0043 \end{bmatrix} u(t) \\
y(t) &= \begin{bmatrix} 0 & 0.2500 \end{bmatrix} x(t)
\end{aligned}
$$

where the inputs are elevator deflection, left aileron deflection, and right aileron deflection. The eigenvalues of the open-loop system are $-0.7543 \pm 0.8937i$. The maneuver under consideration is the step change in the pitch-rate. The initial actuator configuration is such that only the desirable input is the elevator input to accomplish this task. The left and right ailerons are considered as redundant actuators and are to be used if elevator fails. For demonstrating reconfiguration capability two different cases are considered: Case (i) system reaches the steady-state condition and elevator fails to hold its position and elevator input suddenly drops to zero; and Case (ii) before system reaches to target output elevator freezes in its position and some constant input is acting on the system but not adequate to reach the desired position. These two cases are chosen as they represent two quite different dynamic characteristics. In the first case the redundant actuators (both ailerons) have to compensate for no elevator input and try to return system to desired steady-state condition whereas in the second case the ailerons have to make-up for the inadequate control input to reach to the desired steady-state. Given below are reconfiguration results based on the methodology presented in Section 5.

Reconfiguration strategy and GPC control law design: For both cases, the steps involved in the control law design are same and are given below. The reconfiguration matrix Q_{rc} has its initial settings as:

$$
Q_{rc} = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix},
$$

The outer product of Q_{rc} with the system transfer matrix yields the following system

$$\dot{x}(t) = \begin{bmatrix} 0 & -1.3677 \\ 1.0000 & -1.5087 \end{bmatrix} x(t) + \begin{bmatrix} 0.2500 \\ 0.2758 \end{bmatrix} u(t)$$

$$y(t) = \begin{bmatrix} -0.0128 & -0.0665 \end{bmatrix} x(t) \tag{57}$$

The discretization of the system of Eq. (57) with sampling time, $T = 0.05s$, we get

$$x(k+1) = \begin{bmatrix} 0.9983 & -0.0658 \\ 0.0481 & 0.9257 \end{bmatrix} x(k) + \begin{bmatrix} 0.0120 \\ 0.0136 \end{bmatrix} u(k)$$

$$y(k) = \begin{bmatrix} -0.0128 & -0.0665 \end{bmatrix} x(k) \tag{58}$$

Now, using the transformation given in the Appendix B we can transform the system to a new system which has $\Delta u(k)$ as input as follows:

$$x(k+1) = \begin{bmatrix} 0.9983 & -0.0658 & 0.0120 \\ 0.0481 & 0.9257 & 0.0136 \\ 0 & 0 & 1.0000 \end{bmatrix} x(k) + \begin{bmatrix} 0.0120 \\ 0.0136 \\ 1.0000 \end{bmatrix} \Delta u(k)$$

$$y(k) = \begin{bmatrix} -0.0128 & -0.0665 & 0 \end{bmatrix} x(k) \tag{59}$$

It can be verify that the system in Eq. (59) is stabilizable but not detectable. In order to satisfy the conditions of the stability theorem given in Section 4, we define a fictitious output and augment the output matrix to be

$$\mathbf{C}_{aug} = \begin{bmatrix} -0.0128 & -0.0665 & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{60}$$

With this newly defined $\mathbf{C}$ matrix, the system is stabilizable and detectable. Now, choosing the GPC design parameters as: $N_1 = 1$, $N_2 = N_u = 5$, $\lambda = 0.1$, and end-point weighting matrix $\mathbf{Q} = diag\{150 \ 800 \ 1\}$ we can obtain a stabilizing control law. The Riccati solutions P_0 and P_1 for these set of design parameters are given below.

$$P_0 = \mathbf{Q} + \mathbf{C}^T \mathbf{C}$$
$$= \begin{bmatrix} 150.0002 & 0.0009 & 0 \\ 0.0009 & 800.0044 & 0 \\ 0 & 0 & 1.0000 \end{bmatrix}$$

$$P_1 = \begin{bmatrix} 147.0957 & 7.5878 & 0.1832 \\ 7.5878 & 608.4186 & 0.7829 \\ 0.1832 & 0.7829 & 0.0921 \end{bmatrix}$$

These Riccati solutions satisfy the conditions of Theorem 1 as the eigenvalues of the difference $P_0 - P_1$ are

$$\text{Eig}(P_0 - P_1) = \begin{bmatrix} 191.8935 & 2.6266 & 0.8780 \end{bmatrix} \geq 0 \ \Rightarrow \ P_0 - P_1 \succeq 0.$$

When the system is reconfigured (in cases (i) or (ii)), the reconfiguration matrix Q_{rc} is given by:

$$Q_{rc} = \begin{bmatrix} 0 & 1 & 1 \end{bmatrix}. \tag{61}$$

22

Then, from Eq. (53) the system transfer matrix after reconfiguration effectively becomes

$$
\dot{x}(t) = \begin{bmatrix} 0.0000 & -1.3677 \\ 1.0000 & -1.5087 \end{bmatrix} x(t) + \begin{bmatrix} -0.0234 & -0.0234 \\ -0.0345 & -0.0345 \end{bmatrix} u(t)
$$

$$
y(t) = \begin{bmatrix} -0.0000 & 0.0313 \end{bmatrix} x(t) \tag{62}
$$

Again, discretization of the system with sampling time, $T = 0.05s$, yields

$$
x(k+1) = \begin{bmatrix} 0.9983 & -0.0658 \\ 0.0481 & 0.9257 \end{bmatrix} x(k) + \begin{bmatrix} -0.0011 & -0.0011 \\ -0.0017 & -0.0017 \end{bmatrix} u(k)
$$

$$
y(k) = \begin{bmatrix} -0.0000 & 0.0313 \end{bmatrix} x(k) \tag{63}
$$

Again, as before, using the transformation given in the Appendix B one can transform the above system to a new system with $\Delta u(k)$ as input as follows:

$$
x(k+1) = \begin{bmatrix} 0.9983 & -0.0658 & -0.0011 & -0.0011 \\ 0.0481 & 0.9257 & -0.0017 & -0.0017 \\ 0 & 0 & 1.0000 & 0 \\ 0 & 0 & 0 & 1.0000 \end{bmatrix} x(k) + \begin{bmatrix} -0.0011 & -0.0011 \\ -0.0017 & -0.0017 \\ 1.0000 & 0 \\ 0 & 1.0000 \end{bmatrix} \Delta u(k)
$$

$$
y(k) = \begin{bmatrix} -0.0000 & 0.0313 & 0 & 0 \end{bmatrix} x(k) \tag{64}
$$

This system again is not detectable and one can define new fictitious output as before and define a new $\mathbf{C}$ matrix as

$$
\mathbf{C}_{aug} = \begin{bmatrix} -0.0000 & 0.0313 & 0 & 0 \\ 0 & 0 & 1.0000 & 0 \\ 0 & 0 & 0 & 1.0000 \end{bmatrix} \tag{65}
$$

The modified system with this new $\mathbf{C}$ matrix is again controllable and observable. Now if we define the new GPC parameters to be $N_1 = 1$, $N_2 = N_u = 5$, $\lambda = 0.1$, and the end-point weighting matrix as $\mathbf{Q} = diag\{130 \quad 200 \quad 1 \quad 1\}$, we obtain the following Riccati solutions which satisfy the stability condition of Theorem 1 as shown below:

$$
P_0 = \mathbf{Q} + \mathbf{C}^T \mathbf{C}
$$

$$
= \begin{bmatrix} 130.0000 & -0.0000 & 0 & 0 \\ -0.0000 & 200.0010 & 0 & 0 \\ 0 & 0 & 1.0000 & 0 \\ 0 & 0 & 0 & 1.0000 \end{bmatrix}
$$

$$
P_1 = \begin{bmatrix} 129.9835 & 0.2793 & -0.0146 & -0.0146 \\ 0.2793 & 171.7884 & -0.0275 & -0.0275 \\ -0.0146 & -0.0275 & 0.0909 & 0.0000 \\ -0.0146 & -0.0275 & 0.0000 & 0.0909 \end{bmatrix}
$$

$$
\mathrm{Eig}(P_0 - P_1) = \begin{bmatrix} 28.2154 & 0.0133 & 0.9095 & 0.9091 \end{bmatrix} \geq 0 \Rightarrow P_0 - P_1 \succeq 0.
$$

Thus the conditions in the Theorem 1 are satisfied and the GPC control law (Eq. (26)) stabilizes the system.

Case (i): Consider the first of the two cases mentioned above. In this case, it is assumed that after system output (pitch rate) reaches its steady-state value of 1.15 deg/sec the elevator input abruptly goes to zero (at $t = 2$ secs) (potentially this could be a result of possible failure) and system is reconfigured using Eq. (6).

Figure 7 shows the system response before and after reconfiguration. It can be seen that the system reaches a zero steady-state error within 2 seconds and when elevator input abruptly goes to zero at $t = 2$ secs, the output starts drifting from the desired value. At $t = 2$ sec reconfiguration takes place and ailerons take over and bring back the output back to the desired value rendering zero steady-state error. The input time histories are given in Fig. 6. As shown in the figure, the elevator deflection has abruptly dropped to zero after reaching an approximate steady-state value of -1.8 deg and at the same instant ailerons take over and bring the output back to the desired value. The ailerons reach the steady deflection of about -19 deg. The cost and tracking error plots are given in Figs. 8 and 9, respectively.
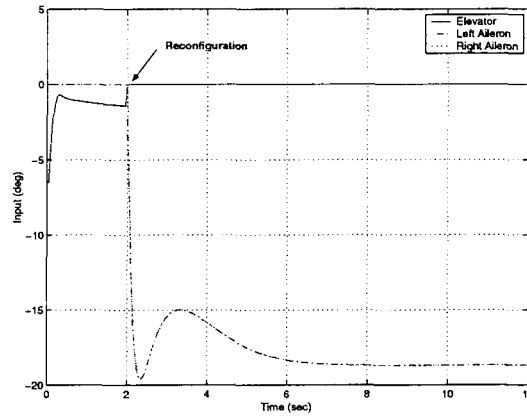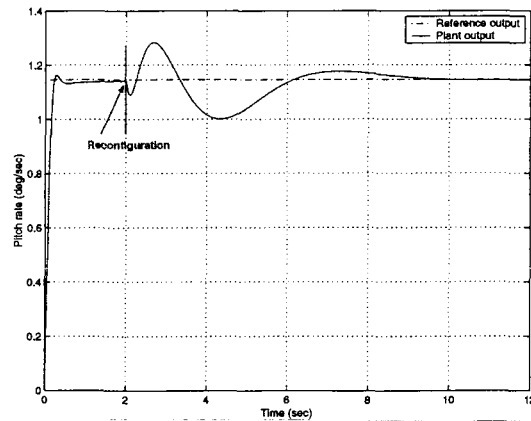
Figure 6: The GPC control law
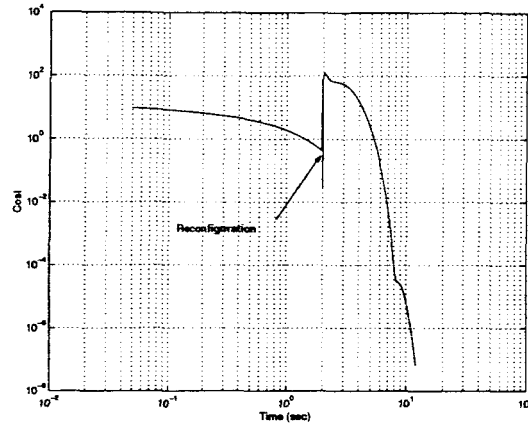
Figure 7: The Plant Output
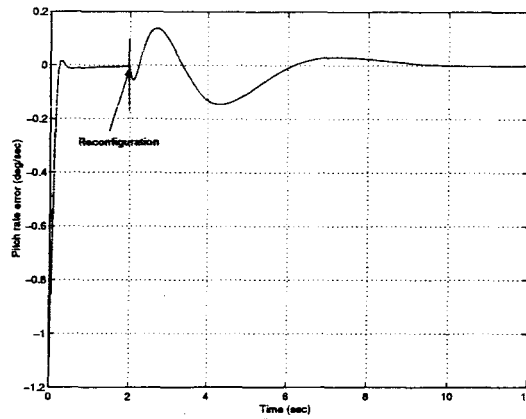
24

Figure 8: The Cost Function

Figure 9: The Plant Output Error

Case (ii): In the second case, it is assumed that the elevator input freezes to some nonzero constant −6 deg before the system output (pitch rate) reaches its desired value (5.73 deg/sec). The simulation results for this case are shown in Figures 10-13. As shown in Fig. 10, the elevator input freezes approximately at −6 deg when the system output has reached only 90% of its final value. At this time ailerons take over as a result of reconfiguration and ailerons deflection reaches a steady-state value of approximately −21 deg as pitch rate reaches to desired value of 5.73 deg/sec. As seen in Fig. 11, overall tracking performance is very satisfactory without steady-state error. The performance function and tracking error time history is shown in Figs. 12 and 13, respectively.

7   Concluding Remarks

This paper presented a comprehensive development of the basic GPC control architecture and derivation of control law for SISO as well as MIMO systems. For SISO systems, both
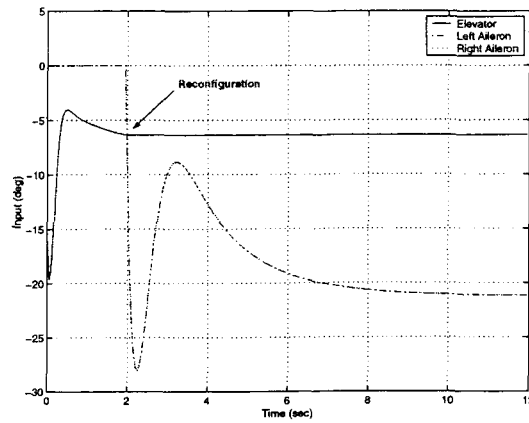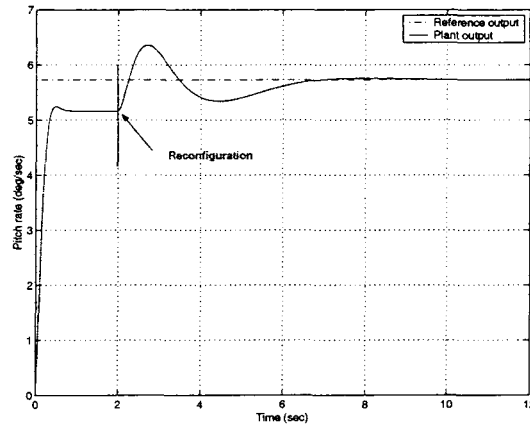
25

Figure 10: The GPC control law

Figure 11: The Plant Output

z-domain (i.e., polynomial-based) as well as state-space based control law development was given. Stability of GPC based on end-point constraints was presented as a basis for establishing stability of a novel reconfigurable GPC control architecture. The stability conditions for reconfiguration scheme were presented. A numerical example consisting of longitudinal control of aircraft was presented to demonstrate the effectiveness of the proposed reconfiguration methodology.
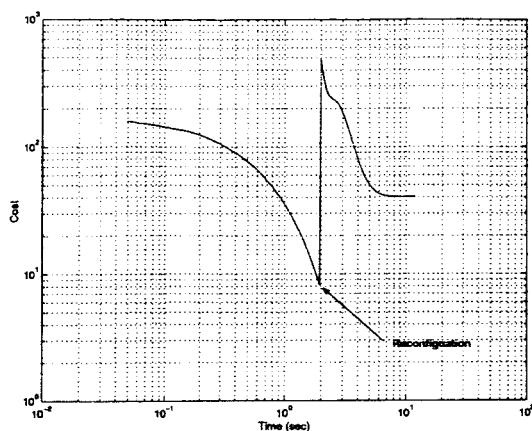
8 Acknowledgements
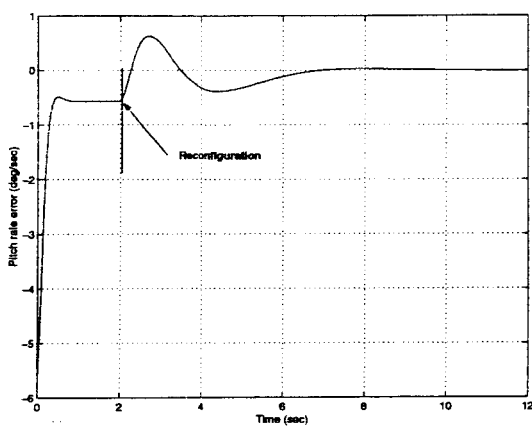
Figure 12: The Cost Function

Figure 13: The Plant Output Error

References

[1] J. Richalet, A. Rault, J.L. Testud, and J. Papon: 'Algorithmic COntrol of Industrial Processes'. In 4th IFAC symposium on Identification and System Parameter Estimation. Tbilisi URSS, 1976.

[2] J. Richalet, A. Rault, J.L. Testud, and J. Papon: 'Model Predictive Heuristic Control: Application to Industrial Processes', Automatica, 14(2):413-428, 1978.

[3] A. I. Propoi: 'Use of LP Methods for Synthesizing Sampled-Data Automatic Systems', Automn Remote COntrol, 24, 1963.

[4] V. Peterka: 'Predictor-Based Self-Tuning Control,' Automatica, 20(1):39-50, 1984.

[5] B. E. Ydstie: 'Extended Horizon Adaptive Control', IN Proc. 9th IFAC World Congress, Budapest, Hungary, 1984.

[6] C. Greco, G. Menga, E. Mosca, and G. Zappa: 'Performance Improvement of Self-tuning COntrollers by Multi-Step Horizons: the MUSMAR approach', Automatica, 20:681-700, 1984.

[7] J. M. Lemos and E. Mosca: 'A Multipredictor-based LQ Self-tuning Controller', In IFAC Symposium on Identification and System Parameter Estimation, York, UK, 137-141, 1985.

[8] J. Richalet, S. Abu el Ata-Doss, C. Arber, H.B. Kuntze, A, Jacubash, and W. Schill: ' Predictive Functional Control: Application to Fast and Accurate Robots,' In Proc. 10th IFAC Congress, Munich, 1987.

[9] R. Soeterboek: 'Predictive Control: A Unified Appraoch,' Prentice Hall, 1992.

[10] M Morari: 'Advances in Model-based Predictive COntrol', Chapter in Model-based Predictive Control: Multivaraible Control Technique of Choice in the 1990s?', Oxford Univ. Press, 1994.

[11] C. Mohtadi: 'Advanced Self-tuning Algorithms', PhD thesis Oxford Univ., U.K., 1986.

[12] D. W. Clarke and Scattolini: 'Constrained Receding-horizon Predictive Control', Proc. IEE, 138(4),:347-354, July 1991.

[13] E. Mosca, J. M. Lemos, and J. Zhang: 'Stabilizing I/O Receding Horizon Control,' IEEE Conf. in Decision and Control, 1990.

[14] K.R. Muske and J. Rawlings: 'Model Predictive Control with Linear Models', AICHE Journal, 39:262-287, 1993.

[15] Ming Ding: 'Neural Network-Based Predictive Control of Uncertain Systems,' M.S. Thesis, Mechanical Engineering Dept., Kansas State University, 1998.

[16] D.W.Clarke, C.Mohtadi, and P.S.Tuffs: Generalized Predictive Control- Part I. The Basic Algorithm, Automatica, 23(2), 1987.

[17] D.W.Clarke, C.Mohtadi, and P.S.Tuffs: Generalized Predictive Control- Part II. Extensions and Interpretations, Automatica, 23(2), 1987.

[18] Properties of Generalized Predictive Control, D.W.Clarke, C.Mohtadi, Automatica, 25(6),1989.

[19] R.R.Bitmead, M.Gevers, V.Wertz, and K.Ogata: Adaptive Optimal Control: The Thinking Man's GPC, Prentice Hall, Englewood Cliffs, New Jersey, 07632, 1990.

[20] H.Demircioglu and D.W.Clarke: Generalized predictive control with end-point state weighting, IEE Proceedings-D, Vol. 140, No.4, July 1993.

[21] C.L.Phillips and H.T.Nagle: Digital Control System Analysis and Design, Third Edition, Prentice Hall, Saddle River, New Jersey 07458, 1995.

[22] E.Mosca: Optimal, Predictive and Adaptive Control, Prentice Hall, Englewood Cliffs, New Jersey, 07632, 1995.

[23] K.Ogata: Discrete-Time Control Systems, Second Edition, Prentice Hall, Englewood Cliffs, New Jersey, 07632, 1995.

[24] E.F.Camacho and C.Bordons: Model Predictive Control, Springer-Verlag London Limited,1999.

[25] Jeffrey B.Burl: Linear Optimal Control: $\mathcal{H}_\in$ and $\mathcal{H}_\infty$ Methods, Addison-Wesley, 1999.

[26] B.L.Stenvens and F.L.Lewis: Aircraft Control and Simulation, John Wiley and Sons, INC. 1992.

[27] R.A.Hess and Y.C.Jung: An Application of Generalized Predictive Control to Rotorcraft Terrain-Following Flight, IEEE Transaction on Systems, Man and Cybernetics, 19(5), 1989.

[28] E.F.Camacho: Constrained Generalized Predictive Control, IEEE Transaction on Automatic Control, 38(2), 1993.

[29] J.A.Rossiter and B.Kouvaritakis: Constrained Stable Generalized Predictive Control, IEE Proceedings-D, 140(4), 1993.

[30] D.Q.Mayne, J.B.Rawlings, C.V.Rao, and P.O.Scokaert: Constrained Model Predictive Control: Stability and Optimality, Automatica, 36, 789-814, 2000.

[31] J.A.Primbs and V.Nevistić: Feasibility and Stability of Constrained Finite Receding Horizon Control, Automatica, 36(7), 956-971, 2000.

[32] J.A.Primbs and V.Nevistić: Constrained Finite Receding Horizon Linear Quadratic Control, Proceedings of 36th Conference on Decision and Control, 1997.

[33] A.Jadbabaie, J.Primbs, and J.Hauser: Unconstrained Receding Horizon Control with No Terminal Cost, Proceedings of the American Control Conference, 2001.

Appendix

A   Diophantine equation

The Diophantine equation offers a convenient way to obtain a division of two polynomials and has the following form:

$$1 = E_j(q^{-1})\widetilde{A}(q^{-1}) + q^{-j}F_j(q^{-1}) \tag{A.1}$$

where $\widetilde{A}(q^{-1}) = \Delta A(q^{-1})$. The degrees of polynomials E_j and F_j are $j - 1$ and n_a, respectively. This equation will be useful in writing predictions of future outputs. The Diophantine equation has a unique solution, E_j and F_j, given $A(q^{-1})$ and the prediction interval j.

Consider Eq. (2) and multiply it by $E_j\Delta q^j$ to obtain

$$E_j\Delta A(q^{-1})y(k+j) = E_j B(q^{-1})\Delta u(k+j-d-1) + E_j\xi(k+j) \tag{A.2}$$

Now substituting for $E_j A\Delta$ from Eq. ($A.1$) gives

$$y(k+j) = F_j(q^{-1})y(k) + E_j(q^{-1})B(q^{-1})\Delta u(k+j-d-1) + E_j(q^{-1})\xi(k+j) \tag{A.3}$$

As the degree of polynomial $E_j(q^{-1}) = j - 1$ the noise terms in Eq. ($A.3$) are all in the future and therefore can be omitted from the prediction as any better prediction of noise is not possible. The best prediction of $y(k+j)$ is therefore given by

$$\widehat{y}(k+j) = G_j(q^{-1})\Delta u(k+j-d-1) + F_j(q^{-1})y(k) \tag{A.4}$$

where, $G_j(q^{-1}) = E_j(q^{-1})B(q^{-1})$. Since the degrees of E_j and F_j are $j - 1$ and n_a, the jth prediction of output involves the past outputs: $(y(k), y(k-1), \cdots, y(k-n_a))$ and inputs $(\Delta u(k+j-d-1), \Delta u(k+j-d-2), \cdots, \Delta u(k-d-n_b))$. The polynomials E_j and F_j obtained from Eq. (A.1) are given by

$$
\begin{aligned}
E_j(q^{-1}) &= e_{j,0} + e_{j,1}q^{-1} + \cdots + e_{j,j-1}q^{-(j-1)} \\
F_j(q^{-1}) &= f_{j,0} + f_{j,1}q^{-1} + \cdots + f_{j,na}q^{-na}
\end{aligned}
$$

Now, the same Diophantine equation can be used to obtain E_{j+1} and F_{j+1}, where

$$F_{j+1}(q^{-1}) = f_{j+1,0} + f_{j+1,1}q^{-1} + \cdots + f_{j+1,na}q^{-na}$$

and the polynomial E_{j+1} is be given by

$$E_{j+1}(q^{-1}) = E_j(q^{-1}) + e_{j+1,j}q^{-j}$$

with $e_{j+1,j} = f_{j,0}$. The polynomial G_{j+1} can now be obtained recursively as follows

$$
\begin{aligned}
G_{j+1} &= E_{j+1}B = (E_j + f_{j,0}q^{-j})B \\
&= G_j + f_{j,0}q^{-j}B
\end{aligned}
\tag{A.5}
$$

where

$$G_j(q^{-1}) = g_{j,0} + g_{j,1}q^{-1} + g_{j,2}q^{-2} + \cdots \qquad (A.6)$$

From Eqs. (A.5) and (A.6), it can be seen that the first j coefficients of G_{j+1} will be identical to those of G_j and the remaining ones are given by

$$g_{j+1,j+i} = g_{j,j+i} + f_{j,0}b_i \qquad i = 0, \cdots, nb$$

In the case when system has a dead time of d, the output of the system at any time instant k is the result of input applied to the system at time $k - d$. As a result, the horizons N_1 and N_2 should be chosen such that $N_2 \geq N_1 \geq d + 1$.

Now consider the following set of j-step ahead optimal predictions for $j = d+1, \cdots, N_2$:

$$
\begin{aligned}
\widehat{y}(k + d + 1) &= G_{d+1}\Delta u(k) + F_{d+1}y(k) \\
\widehat{y}(k + d + 2) &= G_{d+2}\Delta u(k + 1) + F_{d+2}y(k) \\
&\vdots \\
\widehat{y}(k + N_1) &= G_{N_1}\Delta u(k + N_1 - d - 1) + F_{N_1}y(k) \\
&\vdots \\
\widehat{y}(k + N_2) &= G_{N_2}\Delta u(k + N_2 - d - 1) + F_{N_2}y(k)
\end{aligned}
$$

which can be re-written as:

$$\widehat{\mathcal{Y}} = \mathcal{G}\Delta\mathbf{u} + \mathbf{F}(q^{-1})y(k) + \mathcal{G}'(q^{-1})\Delta u(k - 1) \qquad (A.7)$$

where

$$
\widehat{\mathcal{Y}} = \begin{bmatrix} \widehat{y}(k + d + 1) \\ \widehat{y}(k + d + 2) \\ \vdots \\ \widehat{y}(k + N_1) \\ \vdots \\ \widehat{y}(k + N_2) \end{bmatrix}
\qquad
\Delta\mathbf{u} = \begin{bmatrix} \Delta u(k) \\ \Delta u(k + 1) \\ \vdots \\ \Delta u(k + N_u - 1) \end{bmatrix}
$$

$$
\mathcal{G} = \begin{bmatrix}
g_0 & 0 & 0 & \cdots & 0 \\
g_1 & g_0 & 0 & \cdots & 0 \\
\vdots & \vdots & \vdots & \vdots & \vdots \\
g_{N_1-d-1} & g_{N_1-d-2} & g_{N_1-d-3} & \cdots & 0 \\
\vdots & \vdots & \vdots & \vdots & \vdots \\
g_{N_2-d-1} & g_{N_2-d-2} & g_{N_2-d-3} & \cdots & g_{N_2-N_u-d}
\end{bmatrix}
$$

$$
\mathcal{G}'(q^{-1}) = \begin{bmatrix}
(G_{d+1}(q^{-1}) - g_0)q \\
(G_{d+2}(q^{-1}) - g_0 - g_1q^{-1})q^2 \\
\vdots \\
(G_{N_1-d-1}(q^{-1}) - g_0 - g_1q^{-1} - \cdots - g_{N_1-d-1}q^{-(N_1-d-1)})q^{N_1-d} \\
\vdots \\
(G_{N_2-d-1}(q^{-1}) - g_0 - g_1q^{-1} - \cdots - g_{N_2-d-1}q^{-(N_2-d-1)})q^{N_2-d}
\end{bmatrix}
$$

31

and

$$\mathbf{F}(q^{-1}) = \begin{bmatrix} F_{d+1}(q^{-1}) \\ F_{d+2}(q^{-1}) \\ \vdots \\ F_{d+N_1}(q^{-1}) \\ \vdots \\ F_{d+N_2}(q^{-1}) \end{bmatrix}$$

Thus, the predicted outputs over the prediction horizon are given by

$$\mathbf{y} = \begin{bmatrix} \widehat{y}(k+N_1) \\ \widehat{y}(k+N_1+1) \\ \vdots \\ \widehat{y}(k+N_2) \end{bmatrix} = \mathbf{G}\Delta\mathbf{u} + \mathbf{f} \tag{A.8}$$

where,

$$\mathbf{G} = \begin{bmatrix} g_{N_1-d-1} & g_{N_1-d-2} & \cdots & 0 \\ g_{N_1-d} & g_{N_1-d-1} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ g_{N_2-d-1} & g_{N_2-d-2} & \cdots & g_{N_2-N_u-d} \end{bmatrix}$$

and

$$\begin{aligned} \mathbf{f} &= \begin{bmatrix} F_{d+N_1}(q^{-1}) \\ \vdots \\ F_{d+N_2}(q^{-1}) \end{bmatrix} y(k) \\ &+ \begin{bmatrix} (G_{N_1-d-1}(q^{-1}) - g_0 - g_1 q^{-1} - \cdots - g_{N_1-d-1}q^{-(N_1-d-1)})q^{N_1-d} \\ \vdots \\ (G_{N_2-d-1}(q^{-1}) - g_0 - g_1 q^{-1} - \cdots - g_{N_2-d-1}q^{-(N_2-d-1)})q^{N_2-d} \end{bmatrix} \Delta u(k-1) \end{aligned}$$

A.1 Control law

Using Eq. (A.8), the cost function (Eq.(3)) can be written as

$$J(N_1, N_2, N_u, \lambda) = (\mathbf{G}\Delta\mathbf{u} + \mathbf{f} - \mathbf{w})^T(\mathbf{G}\Delta\mathbf{u} + \mathbf{f} - \mathbf{w}) + \lambda\Delta\mathbf{u}^T\Delta\mathbf{u} \tag{A.9}$$

where

$$\mathbf{w} = \begin{bmatrix} w(t+d+N_1) \\ w(t+d+N_1+1) \\ \vdots \\ w(t+d+N_2) \end{bmatrix}$$

Equation (A.9) can be re-written as

$$J(N_1, N_2, N_u, \lambda) = \frac{1}{2}\Delta\mathbf{u}^T\mathbf{H}\Delta\mathbf{u} + \mathbf{b}\Delta\mathbf{u} + \mathbf{f}_0 \tag{A.10}$$

32

where

$$\begin{aligned}
\mathbf{H} &= 2(\mathbf{G}^T\mathbf{G} + \lambda\mathbf{I}) \\
\mathbf{b} &= 2(\mathbf{f} - \mathbf{w})^T\mathbf{G} \\
\mathbf{f_0} &= (\mathbf{f} - \mathbf{w})^T(\mathbf{f} - \mathbf{w})
\end{aligned}$$

Minimizing cost function (3) to obtain optimal control is an unconstrained optimization problem as no other constraints are considered on control input or plant output. The optimal input $\Delta\mathbf{u}^*$ is obtained by setting the gradient of J with respect to $\Delta\mathbf{u}$ equal to zero. That is,

$$\Delta\mathbf{u}^* = -\mathbf{H}^{-1}\mathbf{b}^T = (\mathbf{G}^T\mathbf{G} + \lambda\mathbf{I})^{-1}\mathbf{G}^T(\mathbf{w} - \mathbf{f}) \tag{A.11}$$

The actual control signal sent to the plant is

$$\Delta u^*(k) = I_1\Delta\mathbf{u}^* \tag{A.12}$$

where, $I_1 = [1,\ 0,\ 0,\dots,0]$ is the $1 \times N_u$ row vector. From Eq. (A.12) it can be seen that $\Delta u^*(k)$ is the first element of vector $\Delta\mathbf{u}^*$.

B System transformation

Typically, most of the system equations are available in the form with the input being u and not Δu. However, the GPC control law developed in Eq. (26) is based on the system having input Δu. This section is intended to provide the equations to transform the system with input u to a system with input Δu.

Consider a MIMO LTI system with p inputs and q outputs.

$$\begin{aligned}
\widehat{\mathbf{x}}(k+1) &= \widehat{\mathbf{A}}\widehat{\mathbf{x}}(k) + \widehat{\mathbf{B}}\mathbf{u}(k) \\
\mathbf{y}(k) &= \widehat{\mathbf{C}}\widehat{\mathbf{x}}(k) + \widehat{\mathbf{D}}\mathbf{u}(k)
\end{aligned} \tag{B.13}$$

where input to the system is $\mathbf{u}$ and output is $\mathbf{y}$. Let $\mathbf{H}(z)$ denote the transfer function from $\mathbf{u}(k)$ to $\mathbf{y}(k)$. Then,

$$\mathbf{Y}(z) = \mathbf{H}(z)\mathbf{U}(z)$$

where $\mathbf{H}(z)$ has a realization

$$\mathbf{H} = \left[\begin{array}{c|c} \widehat{\mathbf{A}} & \widehat{\mathbf{B}} \\ \hline \widehat{\mathbf{C}} & \widehat{\mathbf{D}} \end{array}\right] \tag{B.14}$$

Now let $\mathbf{T}(z)$ be the transfer function from new input $\Delta\mathbf{u}(k)$ to $\mathbf{u}(k)$, i. e.,

$$\mathbf{U}(z) = \mathbf{T}(z)\Delta\mathbf{U}(z)$$

where, $\mathbf{T}(z)$ is $p \times p$ transfer matrix and has the form:

$$\mathbf{T}(z) = \begin{bmatrix} \frac{1}{1-z^{-1}} & & \\ & \ddots & \\ & & \frac{1}{1-z^{-1}} \end{bmatrix}_{p \times p}$$

Suppose $\mathbf{T}(z)$ has a realization

$$\mathbf{T} = \left[\begin{array}{c|c} \mathbf{A}_\Delta & \mathbf{B}_\Delta \\ \hline \mathbf{C}_\Delta & \mathbf{D}_\Delta \end{array}\right] = \left[\begin{array}{c|c} \mathbf{I}_{p \times p} & \mathbf{I}_{p \times p} \\ \hline \mathbf{I}_{p \times p} & \mathbf{I}_{p \times p} \end{array}\right] \tag{B.15}$$

and the state

$$\mathbf{x}_\Delta(k) = \mathbf{u}(k-1).$$

Now, let $\mathbf{H}_\Delta(z)$ denotes the system with input $\Delta\mathbf{u}(k)$, output $\mathbf{y}(k)$, and having following state-space description.

$$\begin{aligned}
\mathbf{x}(k+1) &= \mathbf{A}\mathbf{x}(k) + \mathbf{B}\Delta\mathbf{u}(k) \\
\mathbf{y}(k) &= \mathbf{C}\mathbf{x}(k) + \mathbf{D}\Delta\mathbf{u}(k)
\end{aligned}$$

Then,

$$\begin{aligned}
\mathbf{Y}(z) &= \mathbf{H}(z)\mathbf{T}(z)\Delta\mathbf{U}(z) \\
&= \mathbf{H}_\Delta(z)\Delta\mathbf{U}(k)
\end{aligned}$$

34

where $\mathbf{H}_\Delta(z)$ has realization

$$\mathbf{H}_\Delta = \left[\begin{array}{c|c} \mathbf{A} & \mathbf{B} \\ \hline \mathbf{C} & \mathbf{D} \end{array} \right] = \left[\begin{array}{cc|c} \widehat{\mathbf{A}} & \widehat{\mathbf{B}}\mathbf{C}_\Delta & \widehat{\mathbf{B}}\mathbf{D}_\Delta \\ \mathbf{0} & \mathbf{A}_\Delta & \mathbf{B}_\Delta \\ \hline \widehat{\mathbf{C}} & \widehat{\mathbf{D}}\mathbf{C}_\Delta & \widehat{\mathbf{D}}\mathbf{D}_\Delta \end{array} \right] \qquad \text{(B.16)}$$

and the new state is

$$\mathbf{x}(k) = \left[\begin{array}{c} \widehat{\mathbf{x}}(k) \\ \mathbf{x}_\Delta(k) \end{array} \right] = \left[\begin{array}{c} \widehat{\mathbf{x}}(k) \\ \mathbf{u}(k-1) \end{array} \right]$$

C   Proof of Lemma 1

We will use the principle of optimality to derive the optimal control law. The principle of optimality is well known and is stated below **The Principle of Optimality**: *An optimal policy has the property that whatever the initial state and the initial decision are, the remaining decisions must constitute an optimal policy with regard to the state resulting from the first decision.*

In the context of our problem, the principle may be stated as follows:

If a closed-loop control $u^*(k) = f(x(k))$ is optimal over the interval $0 \leq k \leq N$, it is also optimal over any subinterval $m \leq k \leq N$, where $0 \leq m \leq N$.

Let us define a set of various costs F_j as follows

$$
\begin{aligned}
F_N &= \mathbf{x}^T(k+N)(\mathbf{Q} + \mathbf{C}^T\mathbf{C})\mathbf{x}(k+N) \\
F_j &= \mathbf{x}^T(k+j)\mathbf{C}^T\mathbf{C}\mathbf{x}(k+j) + \lambda\Delta\mathbf{u}^T(k+j)\Delta\mathbf{u}(k+j) \qquad (j = 0, 1, \cdots, N-1)
\end{aligned}
$$

Then, we an re-write the cost function, Eq. (22), as follows

$$
J_N = F_0 + F_1 + \cdots + F_{N-1} + F_N - \mathbf{x}^T(k)\mathbf{C}^T\mathbf{C}\mathbf{x}(k)
$$

Note that $\mathbf{x}(k)$ does not affect the optimal control law $\Delta\mathbf{u}^*(j)$, for $j \geq k$.

Let S_m $(m = 1, 2, \cdots, N+1)$ be the cost from $j = k + N - m + 1$ to $j = k + N$. Then S_m can be written as

$$
S_m = J_N - J_{N-m} = F_{N-m+1} + \cdots + F_N
$$

From the Principle of Optimality if J_N is the optimal cost, then so is S_m. By the definition of S_m, we know that

$$
\begin{aligned}
S_{m+1} &= S_m + F_{N-m} \\
&= S_m + \mathbf{x}^T(k+N-m)\mathbf{C}^T\mathbf{C}\mathbf{x}(k+N-m) + \\
&\quad \lambda\Delta\mathbf{u}^T(k+N-m)\Delta\mathbf{u}(k+N-m) \qquad\qquad (C.17)
\end{aligned}
$$

If S_m^* is the optimal cost then we can get the optimal control $\Delta\mathbf{u}^*(k+N-m)$ by replacing S_m by S_m^* in Eq. (C.17) and then setting

$$
\frac{\partial S_{m+1}}{\partial \Delta\mathbf{u}(k+N-m)} = 0 \qquad\qquad (C.18)
$$

We will obtain the expression for the optimal control by the method of induction. Consider S_1. By the assumptions of GPC, $\Delta\mathbf{u}(k+j) = 0$ for $j \geq N$. So

$$
\begin{aligned}
S_1^* &= \mathbf{x}^T(k+N)(\mathbf{Q} + \mathbf{C}^T\mathbf{C})\mathbf{x}(k+N) \\
&= \mathbf{x}^T(k+N)\widehat{P}(k+N)\mathbf{x}(k+N) \qquad\qquad (C.19)
\end{aligned}
$$

where $\widehat{P}$ is defined as

$$
\widehat{P}(k+N) = \mathbf{Q} + \mathbf{C}^T\mathbf{C}
$$

36

From Eq. (C.17), we get

$$
\begin{aligned}
S_2 &= S_1^* + F_{N-1} \\
&= S_1^* + \mathbf{x}^T(k+N-1)\mathbf{C}^T\mathbf{C}\mathbf{x}(k+N-1) + \lambda\Delta\mathbf{u}^T(k+N-1)\Delta\mathbf{u}(k+N-1)
\end{aligned}
$$

Substituting for S_1^* and using state-space equation $\mathbf{x}(k+1) = \mathbf{A}\mathbf{x}(k) + \mathbf{B}\Delta u(k)$, we get

$$
\begin{aligned}
S_2 &= [\mathbf{A}\mathbf{x}(k+N-1) + \mathbf{B}\Delta u(k+N-1)]^T \widehat{P}(k+N)[\mathbf{A}\mathbf{x}(k+N-1) + \mathbf{B}\Delta u(k+N-1)] \\
&\quad + \mathbf{x}^T(k+N-1)\mathbf{C}^T\mathbf{C}\mathbf{x}(k+N-1) + \lambda\Delta\mathbf{u}^T(k+N-1)\Delta\mathbf{u}(k+N-1)
\end{aligned}
$$

Now setting

$$
\frac{\partial S_2}{\partial \Delta\mathbf{u}(k+N-1)} = 0
$$

we get optimal control law

$$
\Delta\mathbf{u}^*(k+N-1) = -K(k+N-1)\mathbf{x}(k+N-1) \tag{C.20}
$$

where

$$
K(k+N-1) = [\mathbf{B}^T(\mathbf{Q}+\mathbf{C}^T\mathbf{C})\mathbf{B} + \lambda\mathbf{I}]^{-1}\mathbf{B}^T(\mathbf{Q}+\mathbf{C}^T\mathbf{C})\mathbf{A}
$$

and

$$
\begin{aligned}
S_2^* &= \mathbf{x}^T(k+N-1) \times \\
&\quad [(\mathbf{A}-\mathbf{B}K(k+N-1))^T(\mathbf{Q}+\mathbf{C}^T\mathbf{C})(\mathbf{A}-\mathbf{B}K(k+N-1)) + \\
&\quad \mathbf{C}^T\mathbf{C} + \lambda K^T(k+N-1)K(k+N-1)] \times \\
&\quad \mathbf{x}(k+N-1) \\
&:= \mathbf{x}^T(k+N-1)\widehat{P}(k+N-1)\mathbf{x}(k+N-1) \tag{C.21}
\end{aligned}
$$

$\widehat{P}(k+N-1)$ is thus given by

$$
\begin{aligned}
\widehat{P}(k+N-1) &= [(\mathbf{A}-\mathbf{B}K(k+N-1))^T(\mathbf{Q}+\mathbf{C}^T\mathbf{C})(\mathbf{A}-\mathbf{B}K(k+N-1)) + \\
&\quad \mathbf{C}^T\mathbf{C} + \lambda K^T(k+N-1)K(k+N-1)] \tag{C.22}
\end{aligned}
$$

From Eqs. (C.19) and (C.21), by induction we can write

$$
S_m^* = \mathbf{x}^T(k+N-m+1)\widehat{P}(k+N-m+1)\mathbf{x}(k+N-m+1)
$$

and

$$
\begin{aligned}
S_{m+1} &= S_m^* + \mathbf{x}^T(k+N-m)\mathbf{C}^T\mathbf{C}\mathbf{x}(k+N-m) + \lambda\Delta\mathbf{u}^T(k+N-m)\Delta\mathbf{u}(k+N-m) \\
&= [\mathbf{A}\mathbf{x}(k+N-m) + \mathbf{B}\Delta\mathbf{u}(k+N-m)]^T \widehat{P}(k+N-m+1) \times \\
&\quad [\mathbf{A}\mathbf{x}(k+N-m) + \mathbf{B}\Delta\mathbf{u}(k+N-m)] \\
&\quad + \mathbf{x}^T(k+N-m)\mathbf{C}^T\mathbf{C}\mathbf{x}(k+N-m) + \lambda\Delta\mathbf{u}^T(k+N-m)\Delta\mathbf{u}(k+N-m)
\end{aligned}
$$

Now, Eq. (C.18) can be used to find the optimal control law as follows

$$
\frac{\partial S_{m+1}}{\partial \Delta\mathbf{u}(k+N-m)} = 0 \quad \Rightarrow \quad \Delta\mathbf{u}^*(k+N-m) = -K(k+N-m)\mathbf{x}(k+N-m) \tag{C.23}
$$

where

$$K(k + N - m) = [\mathbf{B}^T \widehat{P}(k + N - m + 1)\mathbf{B} + \lambda \mathbf{I}]^{-1} \mathbf{B}^T \widehat{P}(k + N - m + 1)\mathbf{A} \qquad (C.24)$$

Substituting for $\Delta \mathbf{u}$ yields

$$
\begin{aligned}
S_{m+1}^* &= \mathbf{x}^T(k + N - m) \times \\
&\quad [(\mathbf{A} - \mathbf{B}K(k + N - m))^T \widehat{P}(k + N - m + 1)(\mathbf{A} - \mathbf{B}K(k + N - m)) + \\
&\quad \mathbf{C}^T\mathbf{C} + \lambda K^T(k + N - m)K(k + N - m)] \times \\
&\quad \mathbf{x}(k + N - m) \\
&:= \mathbf{x}^T(k + N - m)\widehat{P}(k + N - m)\mathbf{x}(k + N - m)
\end{aligned}
$$

where, $\widehat{P}(k + N - m)$ is given by

$$\widehat{P}(k + N - m) = \mathbf{A}^T \widehat{P}(k + N - m + 1)(\mathbf{A} - \mathbf{B}K(k + N - m)) + \mathbf{C}^T\mathbf{C} \qquad (C.25)$$

and the optimal cost is given by

$$
\begin{aligned}
S_{N+1}^* &= \mathbf{x}^T(k)\widehat{P}(k)\mathbf{x}(k) \\
J_N^* &= S_{N+1}^* - \mathbf{x}^T(k)\mathbf{C}^T\mathbf{C}\mathbf{x}(k) \\
&= \mathbf{x}^T(k)(\widehat{P}(k) - \mathbf{C}^T\mathbf{C})\mathbf{x}(k)
\end{aligned}
$$

From Eqs. (C.20), (C.24) and (C.25), the optimal control law $\Delta \mathbf{u}^*(k)$ is given by

$$\Delta \mathbf{u}^*(k) = -K(k)\mathbf{x}(k) \qquad (C.26)$$

where, $K(k)$ is obtained from the following recursive equation

$$K(k + j) = [\mathbf{B}^T \widehat{P}(k + j + 1)\mathbf{B} + \lambda \mathbf{I}]^{-1} \mathbf{B}^T \widehat{P}(k + j + 1)\mathbf{A} \qquad (C.27)$$

and $\widehat{P}(k)$ is the solution of the Riccati Difference Equation (RDE)

$$\widehat{P}(k+j) = \mathbf{A}^T \widehat{P}(k+j+1)\mathbf{A} - \mathbf{A}^T \widehat{P}(k+j+1)\mathbf{B}[\mathbf{B}^T \widehat{P}(k+j+1)\mathbf{B} + \lambda \mathbf{I}]^{-1} \mathbf{B}^T \widehat{P}(k+j+1)\mathbf{A} + \mathbf{C}^T\mathbf{C} \qquad (C.28)$$

with the boundary condition

$$\widehat{P}(k + N) = \mathbf{Q} + \mathbf{C}^T\mathbf{C} \qquad (C.29)$$

∎

D   Proof of Lemma 2

In this section, we will show that the control law in Eq. (C.26) is same as the one in Eq. (26). We will show by manipulating these equations that their right hand side (RHS) and left hand side (LHS) are same.

Pre-multiplying both sides of Eq. (26) by $\left(\mathbf{G}^T \mathbf{G} + \lambda \mathbf{I} + \overline{C}^T \mathbf{Q} \overline{C} \right)$ we get

$$\left(\mathbf{G}^T \mathbf{G} + \lambda \mathbf{I} + \overline{C}^T \mathbf{Q} \overline{C} \right) \Delta \mathbf{u}^* = - \left(\mathbf{G}^T L + \overline{C}^T \mathbf{Q} \mathbf{A}^N \right) \mathbf{x}(k)$$

Now, substituting $\mathbf{G}$, $\overline{C}$, and L, in above equation yields

$$\begin{bmatrix} \mathbf{B}^T \mathbf{C}^T \mathbf{C} \mathbf{B} + \cdots + \mathbf{B}^T (\mathbf{A}^{N-1})^T (\mathbf{Q} + \mathbf{C}^T \mathbf{C})(\mathbf{A}^{N-1})^T \mathbf{B} & \cdots & \mathbf{B}^T (\mathbf{A}^{N-1})^T (\mathbf{Q} + \mathbf{C}^T \mathbf{C}) \mathbf{B} \\ \vdots & \ddots & \vdots \\ \mathbf{B}^T (\mathbf{Q} + \mathbf{C}^T \mathbf{C}) \mathbf{A}^{N-1} \mathbf{B} & \cdots & \mathbf{B}^T (\mathbf{Q} + \mathbf{C}^T \mathbf{C}) \mathbf{B} \end{bmatrix} \Delta \mathbf{u}^*$$

$$+ \lambda \mathbf{I} \Delta \mathbf{u}^* = - \begin{pmatrix} \mathbf{B}^T \mathbf{C}^T \mathbf{C} \mathbf{A} + \mathbf{B}^T \mathbf{A}^T \mathbf{C}^T \mathbf{C} \mathbf{A}^2 + \cdots + \mathbf{B}^T (\mathbf{A}^{N-1})^T (\mathbf{Q} + \mathbf{C}^T \mathbf{C}) \mathbf{A}^N \\ \mathbf{B}^T \mathbf{C}^T \mathbf{C} \mathbf{A}^2 + \cdots + \mathbf{B}^T (\mathbf{A}^{N-1})^T (\mathbf{Q} + \mathbf{C}^T \mathbf{C}) \mathbf{A}^N \\ \vdots \\ \mathbf{B}^T (\mathbf{Q} + \mathbf{C}^T \mathbf{C}) \mathbf{A}^N \end{pmatrix} \mathbf{x}(k) \quad \text{(D.30)}$$

The ith equation is given by

$$\left[\mathbf{B}^T \mathbf{C}^T \mathbf{C} \mathbf{A}^{i-1} \mathbf{B} + \mathbf{B}^T \mathbf{A}^T \mathbf{C}^T \mathbf{C} \mathbf{A}^i \mathbf{B} + \cdots + \mathbf{B}^T (\mathbf{A}^{N-i})^T (\mathbf{Q} + \mathbf{C}^T \mathbf{C}) \mathbf{A}^{N-1} \mathbf{B} \right] \Delta \mathbf{u}^*(k)$$

$$+ \left[\mathbf{B}^T \mathbf{C}^T \mathbf{C} \mathbf{A}^{i-2} \mathbf{B} + \mathbf{B}^T \mathbf{A}^T \mathbf{C}^T \mathbf{C} \mathbf{A}^{i-1} \mathbf{B} + \cdots + \mathbf{B}^T (\mathbf{A}^{N-i})^T (\mathbf{Q} + \mathbf{C}^T \mathbf{C}) \mathbf{A}^{N-2} \mathbf{B} \right] \Delta \mathbf{u}^*(k+1)$$

$$+ \cdots \cdots$$

$$+ \left[\mathbf{B}^T \mathbf{C}^T \mathbf{C} \mathbf{B} + \mathbf{B}^T \mathbf{A}^T \mathbf{C}^T \mathbf{C} \mathbf{A} \mathbf{B} + \cdots + \mathbf{B}^T (\mathbf{A}^{N-i})^T (\mathbf{Q} + \mathbf{C}^T \mathbf{C}) \mathbf{A}^{N-i} \mathbf{B} \right] \Delta \mathbf{u}^*(k+i-1)$$

$$+ \left[\mathbf{B}^T \mathbf{A}^T \mathbf{C}^T \mathbf{C} \mathbf{B} + \mathbf{B}^T (\mathbf{A}^2)^T \mathbf{C}^T \mathbf{C} \mathbf{A} \mathbf{B} + \cdots + \mathbf{B}^T (\mathbf{A}^{N-i})^T (\mathbf{Q} + \mathbf{C}^T \mathbf{C}) \mathbf{A}^{N-i-1} \mathbf{B} \right] \Delta \mathbf{u}^*(k+i)$$

$$+ \cdots \cdots$$

$$+ \left[\mathbf{B}^T (\mathbf{A}^{N-i})^T (\mathbf{Q} + \mathbf{C}^T \mathbf{C}) \mathbf{B} \right] \Delta \mathbf{u}^*(k+N-1)$$

$$= - \left[\mathbf{B}^T \mathbf{C}^T \mathbf{C} \mathbf{A}^i + \mathbf{B}^T \mathbf{A}^T \mathbf{C}^T \mathbf{C} \mathbf{A}^{i+1} + \cdots + \mathbf{B}^T (\mathbf{A}^{N-i})^T (\mathbf{Q} + \mathbf{C}^T \mathbf{C}) \mathbf{A}^N \right] \mathbf{x}(k) \quad \text{(D.31)}$$

Now from Eq. (C.26), we can write

$$\begin{aligned} \Delta \mathbf{u}^*(k+i) &= -K(k+i) \mathbf{x}(k+i) \\ &= - \left[\mathbf{B}^T \widehat{P}(k+i+1) \mathbf{B} + \lambda \mathbf{I} \right]^{-1} \mathbf{B}^T \widehat{P}(k+i+1) \mathbf{A} \mathbf{x}(k+i) \end{aligned}$$

Pre-multiplying above equation by $\left[\mathbf{B}^T \widehat{P}(k+i+1) \mathbf{B} + \lambda \mathbf{I} \right]$ gives

$$\left[\mathbf{B}^T \widehat{P}(k+i+1) \mathbf{B} + \lambda \mathbf{I} \right] \Delta \mathbf{u}^*(k+i) = -\mathbf{B}^T \widehat{P}(k+i+1) \mathbf{A} \mathbf{x}(k+i)$$

$$\ldots (i = 0, 1, 2, \cdots, N-1) \quad \text{(D.32)}$$

Now, using

$$\mathbf{x}(k+i) = \mathbf{A}^i \mathbf{x}(k) + \sum_{j=0}^{i-1} \mathbf{A}^{i-j-1} \mathbf{B} \Delta \mathbf{u}(k+j) \quad \text{(D.33)}$$

$$= \mathbf{A}^i \mathbf{x}(k) + \begin{bmatrix} \mathbf{A}^{i-1}\mathbf{B} & \mathbf{A}^{i-2}\mathbf{B} & \cdots & \mathbf{B} \end{bmatrix} \begin{pmatrix} \Delta\mathbf{u}(k) \\ \Delta\mathbf{u}(k+1) \\ \vdots \\ \Delta\mathbf{u}(k+i-1) \end{pmatrix} \quad (D.34)$$

we can write Eq. (D.32) as

$$\mathbf{B}^T \widehat{P}(k+i+1) \begin{bmatrix} \mathbf{A}^i\mathbf{B} & \mathbf{A}^{i-1}\mathbf{B} & \cdots & \mathbf{A}^{i-1}\mathbf{B} & \mathbf{B} \end{bmatrix} \begin{pmatrix} \Delta\mathbf{u}^*(k) \\ \Delta\mathbf{u}^*(k+1) \\ \vdots \\ \Delta\mathbf{u}^*(k+i) \end{pmatrix}$$

$$+ \lambda\mathbf{I}\Delta\mathbf{u}^*(k+i) = -\mathbf{B}^T\widehat{P}(k+i+1)\mathbf{A}^{i+1}\mathbf{x}(k) \quad (i=0,1,2,\cdots,N-1)$$

Substituting for index i, we get

$$\begin{pmatrix} \mathbf{B}^T\widehat{P}(k+1)\mathbf{B} \\ \mathbf{B}^T\widehat{P}(k+2)\mathbf{AB} & \mathbf{B}^T\widehat{P}(k+2)\mathbf{B} \\ \vdots & \vdots & \ddots \\ \mathbf{B}^T\widehat{P}(k+N)\mathbf{A}^{N-1}\mathbf{B} & \mathbf{B}^T\widehat{P}(k+N)\mathbf{A}^{N-2}\mathbf{B} & \cdots & \mathbf{B}^T\widehat{P}(k+N)\mathbf{B} \end{pmatrix} \underbrace{\begin{pmatrix} \Delta\mathbf{u}^*(k) \\ \Delta\mathbf{u}^*(k+1) \\ \vdots \\ \Delta\mathbf{u}^*(k+N-1) \end{pmatrix}}_{\Delta\mathbf{u}^*}$$

$$+ \lambda\mathbf{I}\Delta\mathbf{u}^* = - \begin{pmatrix} \mathbf{B}^T\widehat{P}(k+1)\mathbf{A} \\ \mathbf{B}^T\widehat{P}(k+2)\mathbf{A}^2 \\ \vdots \\ \mathbf{B}^T\widehat{P}(k+N)\mathbf{A}^N \end{pmatrix} \mathbf{x}(k) \quad (D.35)$$

Note that Eqs. (D.30) and (D.35) have similar structure. Now, if we transform the RHS of Eq. (D.35) to match that of Eq. (D.30) and, as a result, if we get the LHS of both equations to match we have proved the lemma.

So, first transform the RHS of Eq. (D.35) to match that of Eq. (D.30). Consider the ith row in the RHS of Eq. (D.35). By Eq. (C.25), we have

$$\begin{aligned} \widehat{P}(k+i) &= \mathbf{A}^T\widehat{P}(k+i+1)\mathbf{A} + \mathbf{C}^T\mathbf{C} - \mathbf{A}^T\widehat{P}(k+i+1)\mathbf{B}K(k+i) \\ &= \mathbf{A}^T[\mathbf{A}^T\widehat{P}(k+i+2)\mathbf{A} + \mathbf{C}^T\mathbf{C} - \mathbf{A}^T\widehat{P}(k+i+2)\mathbf{B}K(k+i+1)]\mathbf{A} + \mathbf{C}^T\mathbf{C} \\ & \quad - \mathbf{A}^T\widehat{P}(k+i+1)\mathbf{B}K(k+i) \\ & \quad \vdots \\ &= (\mathbf{A}^{N-i})^T\widehat{P}(k+N)\mathbf{A}^{N-i} + (\mathbf{A}^{N-i-1})^T\mathbf{C}^T\mathbf{C}\mathbf{A}^{N-i-1} + (\mathbf{A}^{N-i-2})^T\mathbf{C}^T\mathbf{C}\mathbf{A}^{N-i-2} + \\ & \quad \cdots + \mathbf{C}^T\mathbf{C} - (\mathbf{A}^{N-i})^T\widehat{P}(k+N)\mathbf{B}K(k+N-1)\mathbf{A}^{N-i-1} \\ & \quad - (\mathbf{A}^{N-i-1})^T\widehat{P}(k+N-1)\mathbf{B}K(k+N-2)\mathbf{A}^{N-i-2} - \cdots \\ & \quad - \mathbf{A}^T\widehat{P}(k+i+1)\mathbf{B}K(k+i) \end{aligned} \quad (D.36)$$

That is, the ith row of RHS in Eq. (D.35) becomes

$$-\mathbf{B}^T\widehat{P}(k+i)\mathbf{A}^i = -\Phi + \Omega$$

where

$$\Phi = [\mathbf{B}^T(\mathbf{A}^{N-i})^T\widehat{P}(k+N)\mathbf{A}^N + \mathbf{B}^T(\mathbf{A}^{N-i-1})^T\mathbf{C}^T\mathbf{C}\mathbf{A}^{N-1} + \cdots + \mathbf{B}^T\mathbf{C}^T\mathbf{C}\mathbf{A}^i]\mathbf{x}(k)$$

40

$$\text{and } \Omega = [\mathbf{B}^T(\mathbf{A}^{N-i})^T \widehat{P}(k+N)\mathbf{B}K(k+N-1)\mathbf{A}^{N-1} \tag{D.38}$$

$$\mathbf{B}^T(\mathbf{A}^{N-i-1})^T \widehat{P}(k+N-1)\mathbf{B}K(k+N-2)\mathbf{A}^{N-2} - \cdots \tag{D.39}$$

$$\mathbf{B}^T\mathbf{A}^T \widehat{P}(k+i+1)\mathbf{B}K(k+i)\mathbf{A}^i]\mathbf{x}(k) \tag{D.40}$$

Note that, Φ is same as the RHS of Eq. (D.31). Next, we will move Ω to the LHS, and then simplify the equation. Now, by (D.33), we have

$$\mathbf{A}^i\mathbf{x}(k) = \mathbf{x}(k+i) - \sum_{j=0}^{i-1} \mathbf{A}^{i-j-1}\mathbf{B}\Delta\mathbf{u}(k+j) \tag{D.41}$$

$$\tag{D.42}$$

Substituting Eq. (D.41) in Ω and using optimal control law, $\Delta\mathbf{u}^*(k+j) = -K(k+j)\mathbf{x}(k+j)$, we get

$\Omega =$

$-\mathbf{B}^T(\mathbf{A}^{N-i})^T \widehat{P}(k+N)\mathbf{B}\Delta\mathbf{u}^*(k+N-1)$

$-\mathbf{B}^T(\mathbf{A}^{N-i})^T \widehat{P}(k+N)\mathbf{B}K(k+N-1)\left(\mathbf{A}^{N-2}\mathbf{B}\Delta\mathbf{u}(k) + \cdots + \mathbf{B}\Delta\mathbf{u}(k+N-2)\right)$

$-\mathbf{B}^T(\mathbf{A}^{N-i-1})^T \widehat{P}(k+N-1)\mathbf{B}\Delta\mathbf{u}^*(k+N-2)$

$-\mathbf{B}^T(\mathbf{A}^{N-i-1})^T \widehat{P}(k+N-1)\mathbf{B}K(k+N-2)\left(\mathbf{A}^{N-3}\mathbf{B}\Delta\mathbf{u}(k) + \cdots + \mathbf{B}\Delta\mathbf{u}(k+N-3)\right)$

$- \cdots \cdots$

$-\mathbf{B}^T\mathbf{A}^T \widehat{P}(k+i+1)\mathbf{B}\Delta\mathbf{u}^*(k+i)$

$-\mathbf{B}^T\mathbf{A}^T \widehat{P}(k+i+1)\mathbf{B}K(k+i)\left(\mathbf{A}^{i-1}\mathbf{B}\Delta\mathbf{u}(k) + \cdots + \mathbf{B}\Delta\mathbf{u}(k+i-1)\right)$

The the ith row of LHS of Eq. (C.26) is

$$\mathbf{B}^T\widehat{P}(k+i)\mathbf{A}^{i-1}\mathbf{B}\Delta\mathbf{u}^*(k) \quad + \quad \mathbf{B}^T\widehat{P}(k+i)\mathbf{A}^{i-2}\mathbf{B}\Delta\mathbf{u}^*(k+1) \cdots$$
$$+ \quad \mathbf{B}^T\widehat{P}(k+i)\mathbf{B}\Delta\mathbf{u}^*(k+i-1) \tag{D.43}$$

Substituting Eq. (D.43) in Eq. (D.36) and moving Ω to the LHS, we get the coefficients of $\Delta u^*(k)$ on LHS as

$$\mathbf{B}^T[(\mathbf{A}^{N-i})^T \widehat{P}(k+N)\mathbf{A}^{N-i} + (\mathbf{A}^{N-i-1})^T\mathbf{C}^T\mathbf{C}\mathbf{A}^{N-i-1} + \cdots + \mathbf{C}^T\mathbf{C}]\mathbf{A}^{i-1}\mathbf{B}$$

where, $\widehat{P}(k+N) = \mathbf{C}^T\mathbf{C} + \mathbf{Q}$. This coefficient is same as the one in Eq. (D.31). Similarly, we can relate the coefficients of $\Delta u^*(k+j)$ $(j = 1, 2, \cdots, i-1)$ to the ones in Eq. (D.31). Next, look at the coefficients of $\Delta u^*(k+j)$, $(j = i, i+1, \cdots, N-1)$. For $j = N-1$, the coefficient of $\Delta u^*(k+j))$ on LHS becomes $\mathbf{B}^T(\mathbf{A}^{N-i})^T \widehat{P}(k+N)\mathbf{B}$, which is same as in Eq. (D.31). For $j = i$, the coefficient of $\Delta u^*(k+j))$ on LHS is given by

$$\mathbf{B}^T\mathbf{A}^T \widehat{P}(k+i+1)\mathbf{B} + \mathbf{B}^T(\mathbf{A}^2)^T \widehat{P}(k+i+2)\mathbf{B}K(k+i+1)\mathbf{B}$$
$$+ \cdots + \mathbf{B}^T(\mathbf{A}^{N-i-1})^T \widehat{P}(k+N-1)\mathbf{B}K(k+N-2)\mathbf{A}^{N-i-3}\mathbf{B}$$
$$+\mathbf{B}^T(\mathbf{A}^{N-i})^T \widehat{P}(k+N)\mathbf{B}K(k+N-1)\mathbf{A}^{N-i-2}\mathbf{B}$$

From Eq. (D.36), we have

$\widehat{P}(k+i+1)$
$= (\mathbf{A}^{N-i-1})^T \widehat{P}(k+N)\mathbf{A}^{N-i-1} + (\mathbf{A}^{N-i-2})^T\mathbf{C}^T\mathbf{C}\mathbf{A}^{N-i-2} + \cdots + \mathbf{C}^T\mathbf{C}$
$-(\mathbf{A}^{N-i-1})^T \widehat{P}(k+N)\mathbf{B}K(k+N-1)\mathbf{A}^{N-i-2}$
$-(\mathbf{A}^{N-i-2})^T \widehat{P}(k+N-1)\mathbf{B}K(k+N-2)\mathbf{A}^{N-i-3} - \cdots$
$-\mathbf{A}^T \widehat{P}(k+i+2)\mathbf{B}K(k+i+1)$

Substituting $\widehat{P}(k+i+1)$, we get the coefficient of $\Delta \mathbf{u}^*(k+i)$ as follows

$$\mathbf{B}^T(\mathbf{A}^{N-i})^T\widehat{P}(k+N)\mathbf{A}^{N-i-1}\mathbf{B} + \mathbf{B}^T(\mathbf{A}^{N-i-1})^T\mathbf{C}^T\mathbf{C}\mathbf{A}^{N-i-2}\mathbf{B}$$
$$+ \cdots + \mathbf{B}^T\mathbf{A}^T\mathbf{C}^T\mathbf{C}\mathbf{B}$$

which is same as the one in Eq. (D.31).

Thus, we have shown that the optimal control law given by Eq. (26) is same as the one given by Eq. (C.26) ∎

Numerical Example:

Consider an LTI system,

$$\begin{aligned} \mathbf{x}(k+1) &= \mathbf{A}\mathbf{x}(k) + \mathbf{B}\Delta\mathbf{u}(k) \\ \mathbf{y}(k) &= \mathbf{C}\mathbf{x}(k) \end{aligned}$$

where

$$\mathbf{A} = \begin{bmatrix} 1 & 2 \\ 3 & 1 \end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \quad \mathbf{C} = \begin{bmatrix} 1 & 2 \end{bmatrix}$$

and the cost function of Eq. (22). Let the GPC control parameters be $\lambda = 1$, $N_1 = 1$, $N_2 = N_u = 3$, and

$$\mathbf{Q} = [1].$$

Now the GPC control law computed by Eq. (26) yields

$$\begin{aligned} K_{gpc} &= (\mathbf{G}^T\mathbf{G} + \lambda\mathbf{I} + \overline{C}^T\mathbf{Q}\overline{C})^{-1}(\overline{C}\mathbf{Q}\mathbf{A}^3 + \mathbf{G}^T L) \\ &= \begin{bmatrix} 3.6368 & 2.4684 \\ 1.8240 & 3.1017 \\ 0.1072 & -0.3723 \end{bmatrix} \end{aligned}$$

Similarly, the LQ control law computation for the same cost function using Eq. (D.35) yields

$$\begin{aligned} K_{lqr} &= \begin{bmatrix} \mathbf{B}^T\widehat{P}(1)\mathbf{B}+\lambda & & \\ \mathbf{B}^T\widehat{P}(2)\mathbf{A}\mathbf{B} & \mathbf{B}^T\widehat{P}(2)\mathbf{B}+\lambda & \\ \mathbf{B}^T\widehat{P}(3)\mathbf{A}^2\mathbf{B} & \mathbf{B}^T\widehat{P}(3)\mathbf{A}\mathbf{B} & \mathbf{B}^T\widehat{P}(3)\mathbf{B}+\lambda \end{bmatrix}^{-1} \begin{pmatrix} \mathbf{B}^T\widehat{P}(1)\mathbf{A} \\ \mathbf{B}^T\widehat{P}(2)\mathbf{A}^2 \\ \mathbf{B}^T\widehat{P}(3)\mathbf{A}^3 \end{pmatrix} \\ &= \begin{bmatrix} 3.6368 & 2.4684 \\ 1.8240 & 3.1017 \\ 0.1072 & -0.3723 \end{bmatrix} \end{aligned}$$

which is same as K_{gpc}. This shows that both control laws are the same at time instant $t = k$. When the cost horizons for GPC and LQ strategies are identical.